

**SCORING SYSTEM OTOMATIS PADA LOMBA MENEMBAK
DENGAN TARGET *SILLHOUETTE* HEWAN MENGGUNAKAN
METODE KLASIFIKASI *NAÏVE BAYES***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Muhamad Taufiq Firmansyah
NIM: 145150301111077



PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

SCORING SYSTEM OTOMATIS PADA LOMBA MENEMBAK DENGAN TARGET
SILHOUETTE HEWAN MENGGUNAKAN METODE KLASIFIKASI NAÏVE BAYES

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :

Muhammad Taufiq Firmansyah

NIM: 145150301111077

Skripsi ini telah diuji dan dinyatakan lulus pada
29 Juni 2018

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II



Rizal Maulana, S.T., M.T., M.Sc.
NIK. 201607 891009 1 001



Mochammad Hannafis Hannafi Ichsan., S.ST., M.T.
NIK. 201405 881229 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Dr. Astoro Kusnawan, S.T., M.T., Ph.D
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Penulis menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan penulis, didalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsurunsur plagiasi, penulis bersedia skripsi ini digugurkan dan gelar akademik yang telah penulis peroleh (Sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 28 Mei 2018



Muhamad Taufiq Firmansyah
NIM : 145150301111077

KATA PENGANTAR

Puja dan puji syukur penulis panjatkan kepada Allah SWT atas berkah, rahmat serta hidayah-Nya, penulis dapat menyelesaikan skripsi ini untuk memperoleh gelar Sarjana Teknik. Adapun judul skripsi yang disusun yakni *"Scoring System Otomatis Pada Lomba Menembak Dengan Target Silhouette Hewan Menggunakan Metode Klasifikasi Naïve Bayes"*.

Dalam menyusun skripsi ini, banyak hambatan yang dihadapi oleh penulis. Penulis menyadari tanpa bantuan serta doa dari berbagai pihak, skripsi ini tidak dapat terselesaikan. Pada kesempatan kali ini, penghargaan dan terimakasih yang sebesar-besarnya penulis sampaikan kepada :

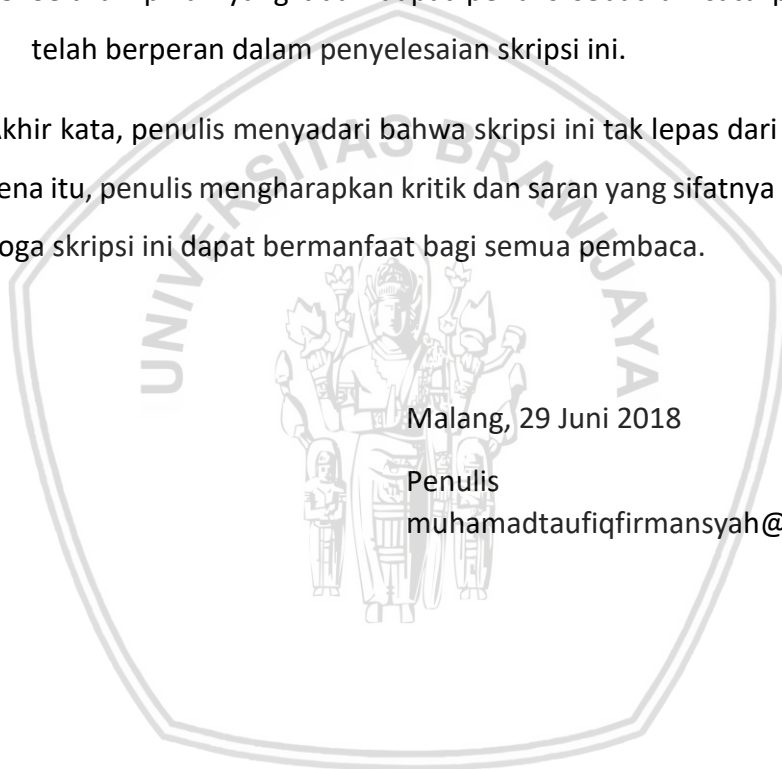
1. Bapak Priyadi Kadarisman dan Ibu Widiati selaku orang tua penulis serta keluarga penulis yang telah memberikan doa dan dukungan.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. Selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Heru Nurwasito, Ir., M.Kom. Selaku Wakil Dekan Bidang Akademik Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. Selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
5. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng. Selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang.
6. Bapak Rizal Maulana, S.T.,M.T.,M.Sc. Selaku Dosen Pembimbing I yang telah memberikan pengarahan dalam pembuatan skripsi ini.
7. Bapak Mochammad Hannats Hanafi Ichsan.,S.ST., M.T. Selaku Dosen Pembimbing II yang telah memberikan pengarahan dalam pembuatan laporan skripsi ini.
8. Seluruh dosen dan karyawan Fakultas Ilmu Komputer yang telah memberikan bantuan dan dukungan selama proses penyelesaian skripsi ini.

9. Handoko Ramadhan., S.T. yang telah memberikan pengarahan dan motivasi kepada penulis.
10. Dulur-dulur Padepokan Laknat yang sudah memberi dukungan dan semangat kepada penulis.
11. Nafisa, Gusti Arief Gilang, dan Muhammad Naufal yang selalu memberi motivasi kepada penulis.
12. Dulur-Dulur HIMATEKKOM serta dulur-dulur Teknik Komputer 2014 yang selalu memberi dukungan kepada penulis.
13. Seluruh pihak yang tidak dapat penulis sebutkan satu persatu yang telah berperan dalam penyelesaian skripsi ini.

Akhir kata, penulis menyadari bahwa skripsi ini tak lepas dari kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang sifatnya membangun dan semoga skripsi ini dapat bermanfaat bagi semua pembaca.

Malang, 29 Juni 2018

Penulis
muhamadtaufiqfirmansyah@gmail.com



ABSTRAK

Menembak adalah olahraga kompetitif yang menguji kemahiran, akurasi dan kecepatan dengan menggunakan berbagai jenis senjata seperti senjata api hingga senjata angin. Perlombaan menembak biasanya diselegarakan oleh Persatuan Penembak Indonesia atau PERBAKIN sebagai bentuk pelatihan serta silaturahmi para anggotanya. Pada perlombaan ini, sistem perhitungan skor masih menggunakan metode manual. Pada penelitian ini akan dibuat *Scoring System* Otomatis Pada Lomba Menembak Dengan Target *Sillhouette* Hewan Menggunakan Metode Klasifikasi *Naïve Bayes* untuk menentukan jenis *Sillhouette* mana yang tertembak dan menambahkan skor pada monitor sistem. Metode Klasifikasi *Naïve Bayes* digunakan dalam proses klasifikasi jenis *Sillhouette* yang tertembak untuk memberi pengaruh terhadap skor dalam perlombaan. Purwarupa pada sistem ini menggunakan beberapa komponen yaitu sensor *ADXL335 Accelerometer*, *Load Cell*, *Limit Switch*, *Switch*, Modul *HX711*, serta *Arduino Mega 2560* dan *Arduino Uno* sebagai pemrosesnya. Berdasarkan hasil uji coba pada sistem ini, didapatkan tingkat akurasi terhadap data sensor sebesar 95,73% dan tingkat akurasi terhadap klasifikasi *Naïve Bayes* adalah sebesar 100%.

Kata Kunci : *Scoring System* Otomatis, Klasifikasi *Naïve Bayes*, *Sillhouette* Hewan

ABSTRACT

Shooting is a competitive sport that tests proficiency, accuracy and speed by using different types of weapons such as firearms to wind weapons. Shooting competitions are usually held by the Indonesian Shooter Union or PERBAKIN as a form of training and gathering of its members. In this competitions, the score calculation system still using manual method. In this research will be made "Automatic Scoring System On Shooting Competition With Target Animals Sillhouette Using the Naïve Bayes Classification Method" to determine which type of Sillhouette is shot and add a score on the system monitor. The Naïve Bayes Classification Method is used in the classification process of the hit Sillhouette to influence the score in the Competition. The prototypes of this system use some components of ADXL335 Accelerometer, Load Cell, Limit Switch, Switch, HX711 Module, and Arduino Mega 2560 and Arduino Uno as processors. Based on the results of the test on this system, the accuracy level of the sensor data is 95.73% and the accuracy of the Naïve Bayes classification is 100%.

Keywords: *Automatic Scoring System, Naïve Bayes Classification, Animal Sillhouette*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 RUMUSAN MASALAH	3
1.3 TUJUAN PENELITIAN	3
1.4 MANFAAT PENELITIAN	4
1.5 BATASAN PENELITIAN	4
1.6 SISTEMATIKA PEMBAHASAN	4
BAB 2 LANDASAN KEPUSTAKAAN	7
2.1 TINJAUAN PUSTAKA	7
2.2 DASAR TEORI	11
2.2.1 <i>Scoring System</i>	11
2.2.2 Menembak	11
2.2.3 <i>Silhouette Hewan</i>	12
2.2.4 <i>Sensor ADXL335 Accelerometer</i>	12
2.2.5 <i>Load Cell</i>	14
2.2.6 <i>Modul HX711</i>	14
2.2.7 <i>Limit Switch</i>	15
2.2.8 <i>Arduino Mega 2560</i>	16
2.2.9 <i>Arduino Uno</i>	17
2.2.10 <i>Analog to Digital Conversion (ADC)</i>	18
2.2.11 <i>Klasifikasi Naïve Bayes</i>	19
BAB 3 METODE PENELITIAN	24
3.1 METODE PENELITIAN	24
3.1.1 Studi Literatur	25

3.1.2 Analisis Kebutuhan	25
3.1.3 Pengambilan Data.....	27
3.1.4 Desain Sistem	27
3.1.5 Implementasi.....	29
3.1.6 Pengujian dan Analisis Hasil	29
3.1.7 Penarikan Kesimpulan dan Saran	29
BAB 4 REKAYASA KEBUTUHAN	31
4.1 Gambaran Umum Sistem	31
4.2 Analisis Kebutuhan Sistem.....	31
4.2.1 Kebutuhan Fungsional	31
4.2.2 Kebutuhan Non-Fungsional	33
4.3 Batasan Desain Sistem.....	34
BAB 5 PERANCANGAN DAN IMPLEMENTASI	36
5.1 Perancangan Sistem	36
5.1.1 Perancangan <i>Prototype Scoring System</i>	36
5.1.2 Perancangan Perangkat Keras	37
5.1.3 Perancangan Perangkat Lunak	41
5.2 Implementasi Sistem	55
5.2.1 Implementasi <i>Prototype Scoring System</i>	56
5.2.2 Implementasi Perangkat Keras	57
5.2.3 Implementasi Perangkat Lunak	59
BAB 6 PENGUJIAN DAN ANALISIS	73
6.1 Pengujian Sensor <i>Load Cell</i>	73
6.1.1 Tujuan Pengujian	73
6.1.2 Prosedur Pengujian.....	73
6.1.3 Hasil dan Analisis Pengujian	75
6.2 Pengujian Sensor <i>ADXL 335 Accelerometer</i>	77
6.2.1 Tujuan Pengujian	77
6.2.2 Prosedur Pengujian.....	77
6.2.3 Hasil dan Analisis Pengujian	78
6.3 Pengujian Sensor <i>Limit Switch</i>	81
6.3.1 Tujuan Pengujian	81
6.3.2 Prosedur Pengujian.....	81
6.3.2 Hasil dan Analisis Pengujian	82

6.4 Pengujian Akurasi Hasil Klasifikasi <i>Naïve Bayes</i>	82
6.4.1 Tujuan Pengujian	82
6.4.2 Prosedur Pengujian.....	83
6.4.3 Hasil dan Analisis Pengujian	85
BAB 7 PENUTUP.....	91
7.1 Kesimpulan	91
7.2 Saran.....	92
DAFTAR PUSTAKA	93
LAMPIRAN.....	94



DAFTAR GAMBAR

Gambar 2.1 <i>Silhouette</i> Hewan	12
Gambar 2.2 MPU-6000 Family Block Diagram.....	13
Gambar 2.3 Sensor MPU6050 Gyroscope & Accelerometer.....	13
Gambar 2.4 <i>Load Cell</i>	14
Gambar 2.5 Modul HX711	15
Gambar 2.6 Limit Switch.....	16
Gambar 2.7 Arduino Mega 2560	16
Gambar 2.8 <i>Arduino Uno</i>	18
Gambar 2.9 Resolusi <i>Sampling</i> ADC	19
Gambar 3.1 <i>Waterfall Diagram</i> Metode Penelitian	24
Gambar 3.2 Analisis Kebutuhan.....	25
Gambar 3.3 Analisis Kebutuhan <i>User</i>	26
Gambar 3.4 Diagram Blok Rancangan Sistem.....	28
Gambar 3.5 Desain Sistem.....	28
Gambar 5.1 <i>Prototype</i> Alat Tampak Samping.....	36
Gambar 5.2 <i>Prototype</i> Alat Tampak Atas	37
Gambar 5.3 Skema Perancangan Perangkat Keras.....	38
Gambar 5.4 Rangkaian <i>Debounce</i>	39
Gambar 5.5 Koneksi <i>I2C</i> antar Arduino	39
Gambar 5.6 <i>Flowchart</i> Perancangan <i>Software</i> pembacaan sensor.....	42
Gambar 5.7 <i>Flowchart</i> Perancangan <i>Software</i> pembacaan sensor <i>Load Cell</i> & <i>Switch</i>	43
Gambar 5.8 <i>Flowchart</i> Perancangan <i>Software</i> pembacaan sensor <i>Accelerometer</i> & <i>Limit Switch</i>	44
Gambar 5.9 <i>Flowchart</i> Perancangan Klasifikasi <i>Naïve Bayes</i>	45
Gambar 5.10 <i>Flowchart</i> Fungsi <i>ProbPrior()</i>	46
Gambar 5.11 <i>Flowchart</i> Fungsi <i>Gaussian()</i>	47
Gambar 5.12 <i>Flowchart</i> Fungsi <i>ProbPosterior()</i>	48
Gambar 5.13 Implementasi <i>Prototype</i> Tampak Depan	56
Gambar 5.14 Implementasi <i>Prototype</i> Tampak Atas.....	57
Gambar 5.15 Implementasi Perangkat Keras pada <i>Box Silhouette</i>	58

Gambar 5.16 Implementasi Perangkat Keras pada <i>Controller</i>	59
Gambar 6.1 <i>Wiring sensor Load Cell</i>	73
Gambar 6.2 <i>Output sensor Load Cell</i>	74
Gambar 6.3 <i>Wiring sensor ADXL 335 Accelerometer</i>	77
Gambar 6.4 <i>Output sensor ADXL 335 Accelerometer</i>	78
Gambar 6.5 <i>Wiring rangkaian Debounce</i>	81
Gambar 6.6 <i>Output Limit Switch</i>	82
Gambar 6.7 <i>Output Klasifikasi Naïve Bayes</i>	83



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	9
Tabel 5.1 Koneksi pin perangkat keras	40
Tabel 5.2 Keterangan dari singkatan masing-masing <i>Silhouette</i>	49
Tabel 5.3 <i>Mean</i> masing-masing <i>Silhouette</i>	51
Tabel 5.4 Standar Deviasi masing-masing <i>Silhouette</i>	52
Tabel 5.5 <i>Gaussian</i> masing-masing <i>Silhouette</i>	54
Tabel 5.6 Sampel data uji masing-masing <i>Silhouette</i>	54
Tabel 5.7 <i>Posterior</i> masing-masing <i>Silhouette</i>	55
Tabel 5.8 Kode program inialisasi library pada sistem	60
Tabel 5.9 Kode program inialisasi variabel pembacaan sensor	60
Tabel 5.10 Kode program pembacaan nilai state push	63
Tabel 5.11 Kode program pembacaan nilai <i>Limit Switch</i>	63
Tabel 5.12 Kode program pembacaan nilai sensor <i>Accelerometer</i>	64
Tabel 5.13 Kode program pembacaan nilai sensor <i>Load Cell</i>	65
Tabel 5.14 Kode program inialisasi variabel dan tipe data metode <i>Naïve Bayes</i>	68
Tabel 5.15 Kode program inialisasi variabel nilai <i>Prior</i> , <i>Mean</i> , & Standar Deviasi	68
Tabel 5.16 Kode program fungsi <i>gaussian</i>	70
Tabel 5.17 Kode program fungsi <i>ProbPosterior</i>	70
Tabel 5.18 Kode program penarikan kesimpulan	71
Tabel 6.1 Hasil Pengujian sensor <i>Load Cell</i>	75
Tabel 6.2 Hasil Pengujian sensor <i>ADXL335 Accelerometer</i>	79
Tabel 6.3 Hasil Pengujian sensor <i>Limit Switch</i>	82
Tabel 6.4 Hasil Pengujian Akurasi Klasifikasi <i>Naïve Bayes</i> Mode Babi 4 Tembakan	85
Tabel 6.5 Hasil Pengujian Akurasi Klasifikasi <i>Naïve Bayes</i> Mode Babi 3 Tembakan A	86
Tabel 6.6 Hasil Pengujian Akurasi Klasifikasi <i>Naïve Bayes</i> Mode Babi 3 Tembakan B	86
Tabel 6.7 Hasil Pengujian Akurasi Klasifikasi <i>Naïve Bayes</i> Mode Ayam 4 Tembakan	86

Tabel 6.8 Hasil Pengujian Akurasi Klasifikasi <i>Naïve Bayes</i> Mode Ayam 3 Tembakan A.....	87
Tabel 6.9 Hasil Pengujian Akurasi Klasifikasi <i>Naïve Bayes</i> Mode Ayam 3 Tembakan B.....	87
Tabel 6.10 Hasil Pengujian Akurasi Klasifikasi <i>Naïve Bayes</i> Mode Kalkun 4 Tembakan	88
Tabel 6.11 Hasil Pengujian Akurasi Klasifikasi <i>Naïve Bayes</i> Mode Kalkun 3 Tembakan A.....	88
Tabel 6.12 Hasil Pengujian Akurasi Klasifikasi <i>Naïve Bayes</i> Mode Kalkun 3 Tembakan B.....	89
Tabel 6.13 Hasil Pengujian Akurasi Klasifikasi <i>Naïve Bayes</i> Mode Domba 4 Tembakan	89
Tabel 6.14 Hasil Pengujian Akurasi Klasifikasi <i>Naïve Bayes</i> Mode Domba 3 Tembakan A.....	89
Tabel 6.15 Hasil Pengujian Akurasi Klasifikasi <i>Naïve Bayes</i> Mode Domba 3 Tembakan B.....	90



DAFTAR LAMPIRAN

Lampiran 1 Peraturan Lomba Menembak Kelas <i>Silhouette</i>	93
Lampiran 2 Data Latih.....	94
Lampiran 3 Kode Program <i>Naïve Bayes</i>	96
Lampiran 4 Kode Program <i>Controller Input</i>	106





BAB 1 PENDAHULUAN

1.1 LATAR BELAKANG

Menembak adalah olahraga kompetitif yang menguji kemahiran, akurasi dan kecepatan dengan menggunakan berbagai jenis senjata seperti senjata api (*Firearms*) hingga senjata angin (*Airguns*) (LenteraKecil, 2016). Olahraga menembak memiliki beberapa jenis cabang, antara lain tembak reaksi, berburu (*Hunting*), tembak sasaran, hingga *Clay Pigeon Shooting*. Dalam olahraga menembak, diperlukan suatu sistem yang dapat menghitung skor dengan akurat sehingga lomba akan menjadi lebih efektif. Biasanya perhitungan skor menembak dilakukan secara manual dengan menggunakan tenaga manusia. *Scoring* bisa dikatakan sebagai suatu metode pemberian harkat terhadap masing-masing value parameter pada perlombaan menembak untuk menentukan nilai yang didapatkan peserta lomba dalam melakukan penembakan terhadap target.

Menembak kelas *Silhouette* merupakan kegiatan menembak sebuah benda mati atau target berupa gambar atau benda yang mirip dengan target hewan yang diburu dalam bentuk mini. Pada umumnya kegiatan ini dilaksanakan pada lomba-lomba menembak. Kelas senapan angin yang dilombakan adalah jenis pompa, per dan gas (pcp), pada umumnya jarak yang dilombakan sejauh 10 meter tanpa menggunakan telescope sedangkan yang menggunakan telescope bisa sejauh 25-40 meter keatas (Ramadhan, Maulana, & Ichsan, 2018). *Silhouette* hewan terdapat banyak varian, seperti ayam, kalkun, babi ataupun domba. Skor pada lomba menembak pun disesuaikan dengan jenis *Silhouette* yang tertembak.

Perlombaan menembak merupakan salah satu olahraga untuk melatih konsentrasi. Pada perlombaan menembak dengan target *Silhouette* yang berbentuk hewan, penggunaan alat Penghitung Skor Otomatis (PSO) masih jarang digunakan (Ramadhan, Maulana, & Ichsan, 2018). Perlombaan ini biasanya diselegarakan oleh Persatuan Penembak Indonesia atau PERBAKIN sebagai bentuk pelatihan serta silaturahmi para anggotanya. Selain itu perlombaan ini juga rutin dilaksanakan oleh pasukan militer diseluruh dunia, termasuk Indonesia. Untuk itu

penulis sangat tertarik untuk melakukan penelitian tentang perlombaan menembak.

Klasifikasi *Naïve Bayes* adalah suatu klasifikasi berpeluang sederhana berdasarkan aplikasi teorema Bayes dengan asumsi antar variabel penjelas saling bebas (independen) (Rivani, Syauqy, & Ichsan, 2017). Dalam hal ini, diasumsikan bahwa kehadiran atau ketiadaan dari suatu kejadian tertentu dari suatu kelompok tidak berhubungan dengan kehadiran atau ketiadaan dari kejadian lainnya. *Naïve Bayes* dapat digunakan untuk berbagai macam keperluan antara lain untuk klasifikasi dokumen, deteksi spam atau filtering spam, dan masalah klasifikasi lainnya. Dalam hal ini lebih disorot mengenai penggunaan teorema *Naive Bayesian* untuk spam *filtering*.

Pada penelitian sebelumnya, sistem serupa menggunakan sensor getar sebagai penentu *Sillhouette* yang terjatuh. Sensor getar memiliki orientasi pada getaran dan hal ini menjadi kelemahan bagi sistem, karena sistem tidak mengetahui getaran seperti apa yang di alami oleh *Sillhouette*. Sistem ini akan lebih cocok jika menggunakan sensor *ADXL335 Accelerometer*. Hal ini dikarenakan sensor *ADXL335 Accelerometer* akan menangkap pergerakan ketika *Sillhouette* terjatuh lebih akurat dibandingkan sensor getar. Selain itu, nilai yang didapat sensor *ADXL335 Accelerometer* lebih cocok untuk diolah pada metode yang akan digunakan pada penelitian ini. Dari segi daya, sensor ini dinilai hemat daya karna sifatnya yang low power (Analog Device, 2010). Pada sistem ini, peneliti menggunakan sensor *ADXL335 Accelerometer* yang berfungsi untuk menangkap pergerakan pada saat tembakan mengenai target.

Dalam penentuan skor, diperlukan metode yang tepat untuk melakukan klasifikasi. Metode *Naive Bayes* merupakan metode klasifikasi yang sangat efektif dan efisien karena dalam melakukan klasifikasi Naive Bayes dapat bekerja secara independen pada setiap fitur-fitur objek yang akan dilakukan klasifikasi. Metode ini hanya memerlukan sejumlah kecil data pelatihan untuk mengestimasi parameter (rata – rata dan variansi dari variabel) yang dibutuhkan untuk klasifikasi. Berdasarkan hal tersebut, penulis bermaksud melakukan penelitian

yang berjudul “*Scoring System* Pada Lomba Menembak Dengan Target *Sillhouette* Hewan Menggunakan Metode *Naïve Bayes*”. Sehingga dengan adanya penelitian ini, diharapkan dapat mempermudah proses penentuan skor pada lomba menembak. Penelitian ini merupakan lanjutan dari penelitian sebelumnya yang berjudul “*Scoring System* Pada Lomba Menembak Dengan Target *Sillhouette* Hewan Menggunakan Logika Fuzzy”. Perbedaan pada penelitian kali ini ialah mengganti sensor getar dengan sensor accelerometer, mengganti penggunaan logika fuzzy dengan metode klasifikasi *naïve bayes*, dan menambahkan fitur pemilihan mode tembak.

1.2 RUMUSAN MASALAH

Dengan permasalahan yang dihadapi, penulis melakukan perumusan masalah sebagai berikut :

1. Bagaimana merancang sistem *scoring* otomatis pada perlombaan menembak dengan target *Sillhouette* hewan?
2. Bagaimana mengimplemetasikan *Naïve Bayes* terhadap sistem *scoring* otomatis pada perlombaan menembak dengan target *Sillhouette* hewan?
3. Bagaimana analisis hasil *Naïve Bayes* dari sistem *scoring* otomatis pada perlombaan menembak dengan target *Sillhouette* hewan?

1.3 TUJUAN PENELITIAN

Dengan dirumuskannya masalah, didapatkan tujuan yang terkait pada rumusan masalah sebagai berikut :

1. Didapatkan perancangan sistem *scoring* otomatis pada perlombaan menembak dengan target *Sillhouette* hewan sehingga dapat diterapkan.
2. Didapat penerapan *Naïve Bayes* terhadap sistem *scoring* otomatis pada perlombaan menembak dengan target *Sillhouette* hewan sehingga didapatkan kesesuaian.
3. Didapatkan kesesuaian *Naïve Bayes* terhadap sistem *scoring* otomatis pada perlombaan menembak dengan target *Sillhouette* hewan sehingga didapatkan hasil yang memrepresentasikan tingkat kesesuaian sistem.

1.4 MANFAAT PENELITIAN

Dilihat dari fungsi sistem yang akan diteliti penulis, didapatkan manfaat sebagai berikut :

1. Dapat membantu orang dalam mengikuti/mengadakan lomba menembak.
2. Kepuasan tersendiri bagi pembuat karena bisa membantu orang lain melalui sistem otomatis perhitungan skor dalam lomba menembak.

1.5 BATASAN PENELITIAN

Adapun agar pembahasan menjadi lebih terarah dan tidak menyimpang sesuai dengan latar belakang yang telah diuraikan. Oleh karena itu penulis melakukan pembatasan penelitian yang dapat dicantumkan sebagai berikut :

1. Target berupa *Sillhouette* yang berbentuk seperti hewan sejumlah 4 *Sillhouette*.
2. Monitor yang akan digunakan sebagai informasi nilai skor pada perlombaan menembak.
3. Sensor *ADXL 335 Accelerometer* dan *Load Cell* dikalibrasi secara manual.

1.6 SISTEMATIKA PEMBAHASAN

Guna memahami lebih jelas pembahasan skripsi ini, dilakukan dengan cara mengelompokkan materi menjadi beberapa sub bab dengan sistematika pembahasan sebagai berikut :

BAB I : PENDAHULUAN

Bab ini menjelaskan tentang informasi umum yaitu latar belakang penelitian, perumusan masalah, tujuan dan manfaat penelitian, batasan penelitian, dan sistematika penelitian tentang "*Scoring System* Pada Lomba Menembak Dengan Target *Sillhouette* Hewan Menggunakan Metode *Naïve Bayes*".

BAB II : LANDASAN TEORI

Bab ini berisikan teori yang diambil dari beberapa kutipan buku, jurnal maupun web yang dibuat oleh lembaga resmi yang bisa dipertanggung jawabkan yang berupa pengertian dan definisi. Bab ini juga menjelaskan konsep dasar metode klasifikasi *Naïve Bayes*, sensor *ADXL335 Accelerometer* pada sistem alat, dan informasi lain yang berkaitan dengan batasan masalah yang peneliti berikan.

BAB III : METODE PENELITIAN

Bab ini berisikan langkah-langkah yang harus dilakukan pada penelitian, yaitu :

1. Studi Literatur
2. Analisis Kebutuhan
3. Pengambilan Data
4. Desain Sistem
5. Implementasi
6. Pengujian dan Analisis Hasil
7. Pengambilan Kesimpulan dan Saran

BAB IV : REKAYASA KEBUTUHAN

Bab ini membahas mengenai persyaratan yang dibutuhkan dalam membangun *prototype scoring system* otomatis pada lomba menembak dengan target *Silhouette* hewan menggunakan metode klasifikasi *Naïve Bayes*, kebutuhan sistem, *software* dan *hardware*.

BAB V : PERANCANGAN DAN IMPLEMENTASI

Bab ini membahas mengenai perancangan *prototype scoring system* otomatis pada lomba menembak dengan target *Silhouette* hewan menggunakan metode klasifikasi *Naïve Bayes*.

BAB VI : PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai pengujian dan analisa terhadap *prototype scoring system* otomatis pada lomba menembak dengan target *Sillhouette* hewan menggunakan metode klasifikasi *Naïve Bayes*.

BAB VII : PENUTUP

Bab ini memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat keras dan lunak yang dikembangkan dalam skripsi ini serta saran-saran untuk pengembangan lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi landasan kepustakaan yang meliputi kajian pustaka dan dasar teori yang diperlukan untuk penelitian ini. Kajian pustaka membahas penelitian sebelumnya yang terkait dengan penelitian yang diusulkan. Adapun dasar teori membahas terkait literatur yang diperlukan untuk menyusun penelitian yang diusulkan.

2.1 TINJAUAN PUSTAKA

Pada penelitian sebelumnya yang telah dibuat oleh Handoko Ramadhan dengan judul *Scoring System Otomatis pada Lomba Menembak dengan Target Silhouette Hewan Menggunakan Logika Fuzzy*. Logika fuzzy digunakan pada proses klasifikasi jenis *Silhouette* untuk mempengaruhi skor pada perlombaan. *Prototype* ini dibangun menggunakan beberapa modul yaitu *Load Cell*, *Sensor Getar*, *Limit Switch*, *Modul HX711*, dan menggunakan *Arduino Uno* serta *Arduino Mega 2560* sebagai pemrosesnya. Berdasarkan hasil uji coba menembak dengan *Silhouette* berbentuk hewan, dibandingkan dengan cara manual yang dilakukan oleh petugas, sistem skoring ini memberikan hasil yang baik. Tingkat akurasi sistem ini sebesar 94.97% pada akurasi data sensornya. Sistem juga dapat mengklasifikasi jenis *Silhouette* menggunakan logika fuzzy dengan tingkat akurasi sebesar 100% (Ramadhan, Maulana, & Ichsan, 2018).

Pada tahun 2017 Prateek Gupta telah merancang sebuah sistem dengan judul *Gesture Controlled Mobile Robotic Arm Vehicle Using Accelerometer* atau model untuk mengontrol Lengan Robot Melalui *Gesture* Menggunakan *Accelerometer*. *Accelerometer* adalah perangkat tiga sumbu yang dipasang pada tangan manusia untuk melakukan aksi lengan robot sesuai tindakan yang dilakukan oleh tangan manusia. Mikrokontroler yang digunakan adalah mikrokontroler PIC, diprogram untuk mengambil pembacaan analog sebagai masukan dari akselerometer dan untuk mentransmisikan sinyal pada ujung penerima lengan robot. Motor DC digunakan untuk mencapai pergerakan lengan

robot, mereka adalah aktuator yang terus berputar seperti motor stepper. Motor ini digunakan dimana ada persyaratan untuk rotasi kontinyu, oleh karena itu digunakan untuk menggerakkan roda. Gripper digunakan untuk memfasilitasi fasilitas pick and drop lengan. Gripper ditempatkan pada platform seluler untuk menyediakan gerakan dari satu tempat ke tempat lain. Tujuan utamanya adalah untuk mengontrol lengan robot melalui gerakan manusia dengan cara berkabel dengan mudah bergerak dalam rentang tertentu (Prateek Gupta, 2017).

Pada tahun 2017 Fauzi Rivani telah penelitian terkait dengan sistem klasifikasi untuk mendeteksi status gizi pada bayi, dengan judul Implementasi *Naive Bayes* pada *Embedded System* untuk Menentukan Status Gizi Bayi. Pada sistem ini membutuhkan parameter-parameter berupa jenis kelamin, umur, panjang dan berat badan bayi. Dalam pembuatan sistem menggunakan komponen elektronika berupa modul, sensor dan mikrokontroler. Untuk memberikan nilai pada parameter jenis kelamin dan umur dibutuhkan modul keypad 4x4 sebagai alat untuk memasukan data, sedangkan untuk nilai dari panjang dan berat badan diakuisisi menggunakan sensor ultrasonik dan *Load Cell*. Kemudian dari data yang telah didapatkan akan diklasifikasi *Arduino Uno* untuk menentukan status gizi dari bayi tersebut. Untuk metode klasifikasi yang digunakan pada penelitian ini adalah *Naive Bayes*. Dari seluruh tahapan awal hingga akhir yang ada pada sistem akan ditampilkan pada modul LCD 16x2. Untuk pengujian dan analisis dilakukan untuk mencari presentase akurasi dari sistem. Dapat disimpulkan untuk pembacaan nilai sensor dari 10 objek, sensor panjang memiliki presentase akurasi 98,28% dan sensor berat 71,02%, pengujian metode klasifikasi dengan 30 data manual secara acak pada sistem status gizi panjang dan berat badan sebesar 100%, kemudian rata-rata dari proses waktu ketika klasifikasi dengan 25 percobaan adalah 0,026 detik dan 0,032 detik. Selanjutnya pengujian dan analisis keseluruhan sistem memiliki presentase akurasi sebesar 96,66% untuk status gizi panjang badan dan 60% untuk status gizi berat badan (Rivani, Syauqy, & Ichsan, 2017).

Muliadi telah dibuat alat *Scoring Board* yang berfungsi untuk mencatat dan menampilkan suatu nilai pencapaian pada suatu *event* olah raga bola basket. Alat

scoring board didesain semi otomatis yang membutuhkan operator untuk melakukan pengendalian. Alat pengontrol berbasis mikrokontroler AT89S51 digunakan agar sistem berjalan semi-otomatis. Alat *scoring board* yang berbasis mikrokontroler ini membutuhkan komponen lainnya sebagai penunjang system yaitu : keypad sebagai *penginput*, decoder 4511 (*latch*) yang menterjemahkan kode-kode biner menjadi bilangan desimal yang ditampilkan melalui *seven segment* yang banyaknya masing-masing 13 buah (Muliadi, 2014).

Berdasarkan empat referensi penelitian yang sebelumnya telah dilakukan, terdapat kemiripan pada objek penelitian mulai dari metode, mikrokontroler, modul dan sistem yang digunakan sehingga penulis memilih empat referensi tersebut sebagai referensi penelitian. Penelitian yang dilakukan memiliki objek penelitian menggunakan perancangan *scoring system* otomatis pada lomba menembak dengan target *Sillhouette* hewan. Sensor yang digunakan pada penelitian penulis menggunakan sensor gyro, *Limit Switch*, *Load Cell*.

Adapun pebandingan kajian pustaka dengan penelitian yang dilusulkan terdapat pada **Tabel 2.1** berikut :

Tabel 2.1 Kajian Pustaka

No.	Nama Penulis [Tahun], Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1.	Handoko Ramadhan [2018], <i>Scoring System</i> Otomatis pada Lomba Menembak dengan Target <i>Sillhouette</i> Hewan Menggunakan Logika <i>Fuzzy</i> .	- Pembuatan alat <i>Scoring</i> System Otomatis pada Lomba Menembak dengan Target <i>Sillhouette</i> Hewan	- Metode yang digunakan adalah Logika <i>Fuzzy</i>	- Metode yang digunakan adalah <i>Naïve</i> <i>Bayes</i>

No.	Nama Penulis [Tahun], Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
		- Sensor yang digunakan untuk mengukur berat <i>Sillhouette</i> adalah sensor <i>Load Cell</i>	- Sensor yang menjadi parameter utama adalah sensor getar	Sensor yang menjadi parameter utama adalah sensor gyro (MPU 6050)
2.	Fauzi Rivani [2017], Implementasi <i>Naïve Bayes</i> pada <i>Embedded System</i> untuk Menentukan Status Gizi Bayi	- Menggunakan metode <i>Naive Bayes</i> dalam proses klasifikasi	- Objek yang diteliti adalah faktor penentuan gizi bayi, antara lain jenis kelamin, panjang badan, dan berat badan.	- Objek yang diteliti adalah faktor penentuan <i>Sillhouette</i> yang tertembak, antara lain berat <i>Sillhouette</i> dan kecepatan kemiringan <i>Sillhouette</i> .
3.	Prateek Gupta [2017], Gesture Controlled Mobile Robotic Arm Vehicle Using <i>Accelerometer</i>	- Menggunakan sensor <i>ADXL335 Accelerometer</i>	- Objek yang diteliti adalah pergerakan robot.	- Objek yang diteliti adalah pergerakan <i>Sillhouette</i> hewan.
4.	Raf Muliadi [2005], Alat <i>Scoring Board</i> Pertandingan Bola Basket Berbasis Mikrokontroler AT89S51	- Membuat <i>Scoring System</i> untuk olahraga	- Diimplementasikan pada pertandingan bola basket	- Diimplementasikan pada lomba menembak

2.2 DASAR TEORI

Sub bab ini berisi dasar teori yang dibutuhkan untuk menyusun penelitian ini.

2.2.1 *Scoring System*

Diagnosis suatu proses pengukuran atribut adalah pemberian makna atau interpretasi terhadap skor skala yang bersangkutan. Sebagai suatu hasil ukur berupa angka (kuantitatif), *scoring system* yang disebut sebagai skor skala, memerlukan suatu norma pembanding agar dapat diinterpretasikan secara kualitatif. Pada dasarnya, interpretasi skor skala selalu bersifat normatif, artinya makna skor diacukan pada posisi relatif skor dalam suatu kelompok yang telah dibatasi terlebih dahulu. Hal ini dapat dilakukan dengan bantuan statistik deskriptif dari distribusi data skor kelompok yang umumnya mencakup banyak subjek (n) dalam kelompok, mean skor skala (M), deviasi standar skor skala (s) dan varians (s^2), skor minimum (x_{min}), maksimum (x_{max}) dan statistik-statistik lain yang dirasa perlu. Deskripsi data ini memberi gambaran penting mengenai keadaan distribusi skor skala pada kelompok subjek yang dikenai pengukuran dan berfungsi sebagai sumber informasi mengenai keadaan subjek pada aspek variabel yang diteliti (Wardhani, 2005).

2.2.2 Menembak

Menembak secara definisi dapat diartikan menembakkan sesuatu pada sesuatu. Olahraga menembak merupakan cabang olahraga yang memerlukan kesinambungan kerja antara aksi dan reaksi dengan baik. Dengan demikian, setiap penembak harus memiliki ketenangan, ketahanan dan pengontrolan diri. Melalui olahraga menembak seseorang dilatih dan dituntut untuk dapat berkonsentrasi tinggi, mampu mengendalikan diri dan berani mengambil keputusan secara cepat dan tepat (LenteraKecil, 2016). Di Indonesia, lomba menembak mempunyai beberapa aturan yang telah ditetapkan oleh Persatuan Penembak Indonesia atau PERBAKIN antara lain :

1. Satu sasaran satu tembakan.

2. Sikap penembak dimulai dari sikap *Prone* (tiarap), Kemudian sikap *Kneeling* (jongkok), dilanjutkan sikap *Standing* (berdiri).
3. Sasaran ditembak dari kiri ke kanan, apabila meleset atau tidak kena tembakan diteruskan ke sasaran berikutnya sampai jumlah tembakan dan waktu habis.

Untuk sistem penilaiannya, tiap satu sasaran yang jatuh diberi nilai 10. Tembakan yang meleset, tidak kena atau kena tapi tidak jatuh diberi nilai 0. Tembakan yang salah sasaran atau tidak sesuai dengan urutan akan dikenakan diskualifikasi pada perlombaan (JEHACESHOOTINGCLUB, 2017).

2.2.3 *Sillhouette* Hewan

Sillhouette hewan adalah sebuah benda mati berbentuk hewan yang biasanya menjadi target dalam lomba menembak. Biasanya benda ini terbuat dari baja agar dapat bertahan ketika terkena tembakan. Bentuk *Sillhouette* yang digunakan sangat berpengaruh dalam lomba menembak, mulai dari jarak sasaran, jarak antar sasaran dan tinggi sasaran (Ramadhan, Maulana, & Ichsan, 2018). Contoh *Sillhouette* hewan terdapat pada Gambar 2.1.



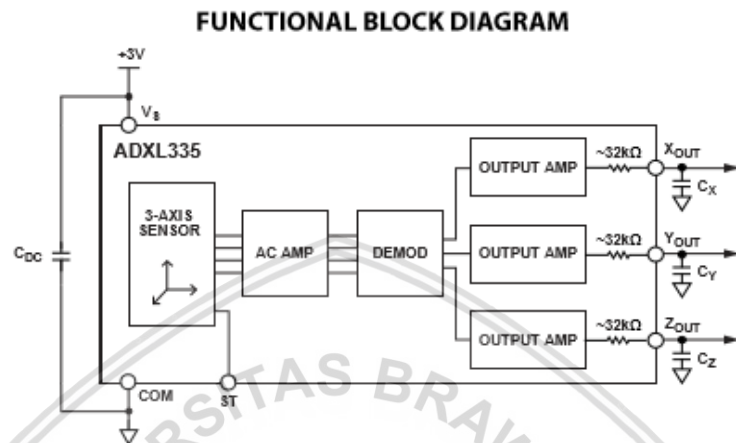
Gambar 2.1 *Sillhouette* Hewan

Sumber : www.gtms shootingclub.com

2.2.4 Sensor *ADXL335 Accelerometer*

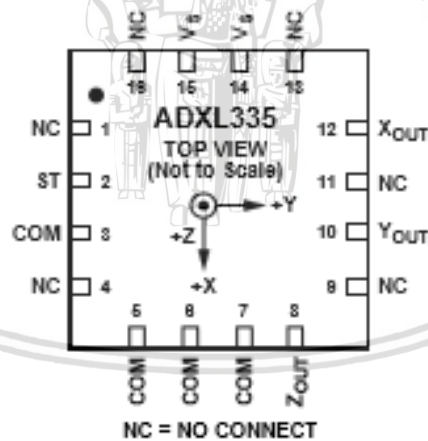
ADXL335 adalah sensor akselerometer 3-sumbu lengkap dengan output tegangan berkategori sinyal, bentuknya tipis, kecil dan berdaya rendah. Sensor ini mengukur akselerasi dengan *minimum full-scale range* ± 3 g. Sensor ini dapat mengukur akselerasi gravitasi statis dalam aplikasi

penginderaan melalui getaran, serta akselerasi dinamis akibat gerak, guncangan, atau getaran (Analog Device, 2010). *Functional Diagram* dan *Pin Configuration* dari sensor ADXL335 tercantum pada Gambar 2.2 dan Gambar 2.3.



Gambar 2.2 MPU-6000 Family Block Diagram

Sumber : www.analog.com

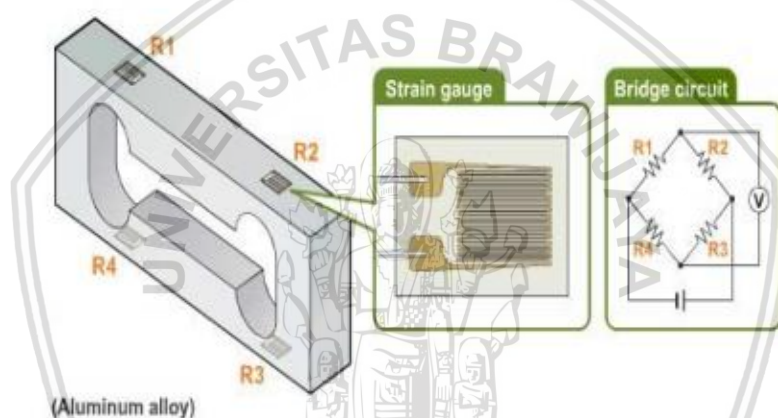


Gambar 2.3 Sensor MPU6050 Gyroscope & Accelerometer

Sumber : www.analog.com

2.2.5 Load Cell

Load Cell adalah sebuah sensor gaya yang banyak digunakan dalam industri yang memerlukan peralatan untuk mengukur berat. Secara umum, *Load Cell* dan sensor gaya berisi pegas (*spring*) logam mekanik dengan mengaplikasikan beberapa *foil metal strain gauges* (SG). *Strain* dari pegas mekanik muncul sebagai pengaruh dari pembebanan yang kemudian ditransmisikan pada *strain gauges*. Pengukuran sinyal yang dihasilkan dari *Load Cell* adalah dari perubahan resistansi *strain gauge* yang linier dengan gaya yang diaplikasikan (Mäuselein, Mack, & Schwartz, 2009). *Detail* dari *Load Cell* terdapat pada Gambar 2.4.

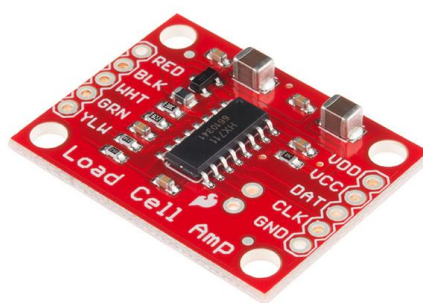


Gambar 2.4 Load Cell

Sumber : www.kitomaindonesia.com

2.2.6 Modul HX711

HX711 merupakan modul timbangan, yang mempunyai prinsip kerja dapat mengkonversi perubahan yang terukur dengan adanya perubahan resistansi dan dapat mengkonversinya ke dalam besaran tegangan melalui rangkaian yang ada. Modul ini memiliki struktur yang sederhana, dan mudah dalam penggunaan, dan mempunyai hasil yang stabil dan *reliable* (Xtopher, 2016). Contoh dari *Modul HX711* terdapat pada Gambar 2.5.



Gambar 2.5 Modul HX711

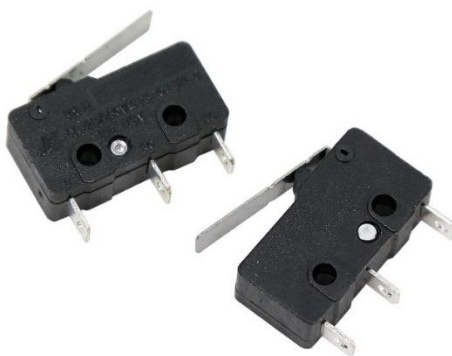
Sumber : sparkfun.com

Spesifikasi modul ini adalah sebagai berikut :

- a) *Differential input voltage* : $\pm 40\text{mV}$ (Full-scale differential input voltage $\pm 40\text{mV}$)
- b) *Data accuracy* : 24 bit (24bit A/D converter chip.)
- c) *Refresh Frequency* : 80 Hz
- d) *Operating Voltage* : 5V DC
- e) *Operating current* : $< 10\text{ mA}$
- f) *Size* : $38\text{mm} \times 21\text{mm} \times 10\text{mm}$

2.2.7 Limit Switch

Limit Switch adalah saklar tekan yang berfungsi sebagai pemutus atau penyambung arus listrik dari sumber arus ke beban listrik. Suatu sistem saklar tekan *Limit Switch* terdiri dari saklar tekan *start*, *stop reset* dan saklar tekan untuk *emergency*. *Limit Switch* memiliki kontak NC (*normally close*) dan NO (*normally open*) (Ramadhan, Maulana, & Ichsan, 2018). Contoh dari *Limit Switch* terdapat pada Gambar 2.6.

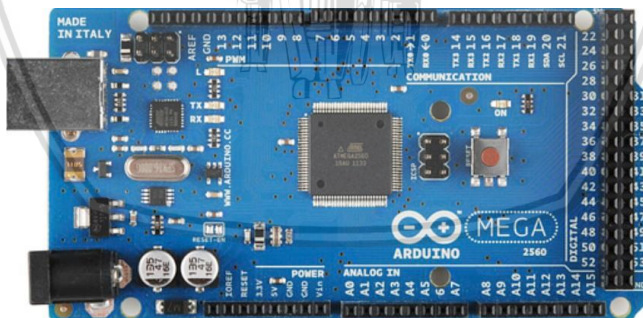


Gambar 2.6 Limit Switch

Sumber : www.robotshop.com

2.2.8 Arduino Mega 2560

Arduino Mega 2560 adalah sebuah mikrokontroller yang berbasis pada ATmega2560 dan merupakan inovasi baru dari board Arduino terdahulunya yaitu *Arduino Mega 2560* dengan *chip* ATmega1280. *Arduino Mega 2560* dapat dihubungkan dengan komputer menggunakan kabel USB dan dapat diprogram dengan mudah karena Arduino cocok dengan bahasa pemrograman C atau C++ yang di *Custom* menjadi sebuah software khusus sebagai *platform* pemrograman untuk Arduino. Contoh dari *board Arduino Mega 2560* terdapat pada Gambar 2.7.



Gambar 2.7 Arduino Mega 2560

Sumber : c-sharpcorner.com

Arduino Mega memiliki keunggulan dari segi *source* penyimpanan yang lebih besar daripada Arduino Uno, selain itu jumlah digital pin yang cukup banyak (54 pin digital) dapat memudahkan user untuk menggunakan banyak komponen sekaligus. Berikut adalah spesifikasi Arduino Mega (Ramadhan, Maulana, & Ichsan, 2018):

1. *Operating Voltage* :5V
2. *Input Voltage (recommended)* : 7-12V
3. *Input Voltage (limits)* : 6-20V
4. *Digital I/O Pins* : 54 (of which 15 provide PWM output)
5. *Analog Input Pins* : 16
6. *DC Current per I/O Pin* : 40 mA
7. *DC Current for 3.3V Pin* : 50 mA
8. *Flash Memory* : 256 KB of which 8 KB used by bootloader
9. *SRAM* : 8 KB
10. *EEPROM* : 4 KB
11. *Clock Speed* : 16 MHz

2.2.9 Arduino Uno

Arduino Uno memiliki 14 pin digital input/output (6 diantaranya dapat digunakan sebagai *output Pulse Width Modulation*), 6 pin analog, sebuah 16MHz kristal osilator, sebuah port USB, sebuah *power jack*, sebuah *ICSP header*, dan sebuah tombol reset (Arduino, 2018). Contoh dari *board Arduino Uno* terdapat pada Gambar 2.8.



Gambar 2.8 Arduino Uno

Sumber : (Arduino, 2018)

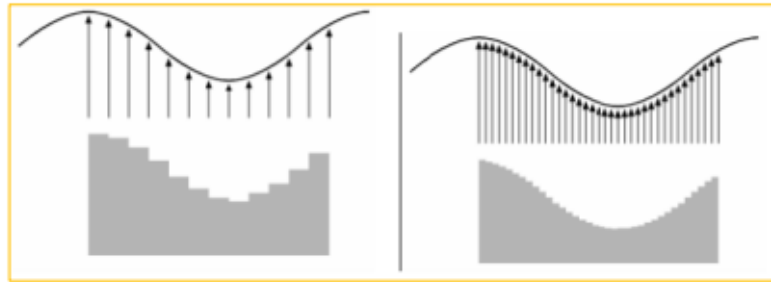
Berikut ini adalah spesifikasi dari *Arduino Uno* beserta deskripsi masing-masing pin (Arduino, 2018) :

1. *Operating Voltage* : 5V
2. *Input Voltage (recommended)* : 7-12V
3. *Input Voltage (limits)* : 6-20V
4. *Digital I/O Pins* : 14 (of which 6 provide PWM output)
5. *Analog Input Pins* : 6
6. *DC Current per I/O Pin* : 20 mA
7. *DC Current for 3.3V Pin* : 50 mA
8. *Flash Memory* : 32 KB of which 0,5 KB used by bootloader
9. *SRAM* : 2 KB
10. *EEPROM* : 1 KB
11. *Clock Speed* : 16 MHz

2.2.10 Analog to Digital Conversion (ADC)

Analog to Digital Converter atau ADC adalah fitur dalam mikrokontroler yang memiliki fungsi untuk mengkonversi (merubah) sinyal masukan analog (dari peranti masukan eksternal) menjadi sinyal masukan digital. Pada mikrokontroler, ADC dapat digunakan untuk berkomunikasi antara mikrokontroler dengan peranti-peranti eksternal (sensor) yang

memiliki gelombang sinyal keluaran berupa gelombang sinyal analog (sinus).
ADC memiliki resolusi *sampling* seperti pada Gambar 2.9.



Gambar 2.9 Resolusi *Sampling* ADC

Sumber : elektronika-dasar.web.id

Resolusi ADC menentukan ketelitian nilai hasil konversi ADC. Sebagai contoh: ADC 8 bit akan memiliki *output* 8 bit data digital, ini berarti sinyal *input* dapat dinyatakan dalam 255 ($2^n - 1$) nilai diskrit. ADC 12 bit memiliki 12 bit *output* data digital, ini berarti sinyal *input* dapat dinyatakan dalam 4096 nilai diskrit. Dari contoh diatas ADC 12 bit akan memberikan ketelitian nilai hasil konversi yang jauh lebih baik daripada ADC 8 bit (Ramadhan, Maulana, & Ichsan, 2018).

$$ADC = \frac{sampel}{max_sampel} \times Range\ ADC \quad (2.1)$$

ADC : Nilai *Analog to Digital Conversion* (ADC)

Sampel : Nilai *output* sampel data analog

Max_sampel : Nilai maksimum sampel data analog

Range ADC: Nilai ADC maksimum

2.2.11 Klasifikasi *Naïve Bayes*

Klasifikasi *Naive Bayes* adalah suatu klasifikasi berpeluang sederhana berdasarkan aplikasi *Teorema Bayes* dengan asumsi antar variabel penjelas saling bebas (independen). Dalam hal ini, diasumsikan bahwa kehadiran atau ketiadaan dari suatu kejadian tertentu dari suatu kelompok tidak berhubungan

dengan kehadiran atau ketiadaan dari kejadian lainnya. Berikut adalah rumus dari *Teorema Bayes* :

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \quad (2.2)$$

Keterangan dari **Persamaan (2.2)** yakni :

$P(y|x)$: Peluang *posterior* (probabilitas kondisional) dari suatu hipotesis kelas y akan terjadi setelah diberikan data x .

$P(x|y)$: Peluang *likelihood* dari sebuah data x terjadi akan mempengaruhi hipotesis kelas y .

$P(y)$: Peluang *prior* (awal) hipotesis kelas y terjadi tanpa memperhatikan data yang diberikan.

$P(x)$: Peluang *evidence* x terjadi tanpa memperhatikan hipotesis kelas/*evidence* lainnya, yakni jumlah total dari semua peluang *likelihood* yang dikalikan dengan peluang *prior*.

Hipotesis dalam *Teorema Bayes* merupakan label kelas yang menjadi target dalam sebuah klasifikasi, sedangkan *evidence* adalah fitur yang menjadi masukan dalam klasifikasi. *Naive Bayes* dilambangkan dengan $P(X|Y)$, dimana X adalah masukan yang berupa fitur-fitur dan Y adalah kelas dalam sebuah klasifikasi. Notasi $P(X|Y)$ berarti peluang kelas Y didapatkan setelah fitur-fitur X diamati, notasi ini merupakan peluang *likelihood* dan $P(Y)$ merupakan notasi dari peluang *prior*. Berikut ini adalah persamaan untuk rumus *Naive Bayes* (Rivani, Syauqy, & Ichsan, 2017):

$$P(Y|X) = \frac{P(Y) \prod_{i=1}^q P(X_i|Y)}{P(X)} \quad (2.3)$$

Keterangan dari **Persamaan (2.3)** yakni :

$P(Y|X)$: Peluang *posterior* (probabilitas kondisional) dari suatu kelas Y akan terjadi setelah mengamati fitur-fitur X.

$\prod_{i=1}^q P(X_i|Y)$: Peluang *likelihood* dari masing-masing fitur X terjadi akan mempengaruhi kelas Y.

$P(Y)$: Peluang *prior* (awal) hipotesis kelas y terjadi tanpa memperhatikan fitur yang diberikan.

$P(X)$: Peluang *evidence* X terjadi tanpa memperhatikan kelas/*evidence* lainnya, yakni jumlah total dari semua peluang *likelihood* yang dikalikan dengan peluang *prior*.

Dalam perhitungan klasifikasi untuk setiap kelas Y yang berbeda akan mempunyai nilai $P(X)$ yang sama, sehingga dalam penentuan klasifikasi *Naive Bayes* selanjutnya ditentukan dari nilai peluang terbesar antara tiap kelas Y dari hasil perhitungan $P(Y) \prod_{i=1}^q P(X_i|Y)$.

Beberapa permasalahan yang ada untuk menentukan nilai peluang dari suatu kondisi yang mudah adalah dengan menghitung peluang dari data diskrit. Namun dalam kenyataannya tidak semua data tersaji dalam bentuk diskrit, tetapi ada yang berbentuk data kontinyu. Untuk itu dalam melakukan proses klasifikasi terhadap data kontinyu dengan *Naive Bayes* terdapat 2 cara, yakni :

1. Melakukan proses perubahan data kontinyu menjadi data diskrit (diskritisasi) terhadap setiap fitur yang akan diestimasi.

2. Menganggap setiap fitur sesuai dengan data latih menggunakan fungsi *univariate normal (Gaussian) distribution* yang ditunjukkan pada **Persamaan (2.4)**, dimana parameter utama dari fungsi *Gaussian* ini adalah *mean* (μ) dan *varian* (σ^2).

$$P(X = x_i | Y = y_i) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}} \quad (2.4)$$

Parameter μ_{ij} bisa didapatkan dari *mean* pada sampel $X_i(\bar{x})$ dari semua data latih yang menjadi milik kelas y_i , sedangkan σ_{ij}^2 dapat dipekirakan dari *varian* sampel (s^2) dari data latih. Adapun fungsi untuk mencari nilai *Mean* dapat dilihat pada **Persamaan (2.5)** berikut :

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (2.5)$$

Perhitungan mean dilakukan dengan menjumlahkan seluruh nilai data suatu kelompok sampel, kemudian dibagi dengan jumlah sampel tersebut. Dimana \bar{x} merupakan rata-rata hitung, x_i merupakan nilai sampel ke-i, dan n adalah jumlah sampel. Adapun fungsi untuk mencari nilai *standar deviasi* dapat dilihat pada **Persamaan (2.6)** berikut :

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \quad (2.6)$$

Untuk menghitung *standar deviasi* yaitu dengan mengurangi setiap nilai data dengan rata-rata kelompok data tersebut, selanjutnya semua hasil dijumlahkan kemudian dibagi dengan jumlah data secara keseluruhan dikurangi 1, dan terakhir hasilnya di akarkan. Dimana s merupakan standar deviasi (simpangan baku), x_i merupakan nilai x ke i, \bar{x} merupakan rata-rata, n adalah ukuran sampel.

2.2.11.1 Alasan Digunakannya Klasifikasi Naïve Bayes

1. Konsep Klasifikasi *Naïve Bayes* sudah familiar dikepala penulis, karena memang sudah diajarkan sebelumnya pada mata kuliah Sistem Pengenalan Pola.
2. Klasifikasi *Naïve Bayes* mudah dipahami.
3. Klasifikasi *Naïve Bayes* menangani kuantitatif dan data diskrit.
4. Klasifikasi *Naïve Bayes* kokoh terhadap atribut yang tidak relevan.
5. Klasifikasi *Naïve Bayes* kokoh untuk titik *noise* yang diisolasi, misalkan titik yang dirata – ratakan ketika mengestimasi peluang bersyarat data.

6. Klasifikasi *Naïve Bayes* hanya memerlukan sejumlah kecil data pelatihan untuk mengestimasi parameter (rata – rata dan variansi dari variabel) yang dibutuhkan untuk klasifikasi.

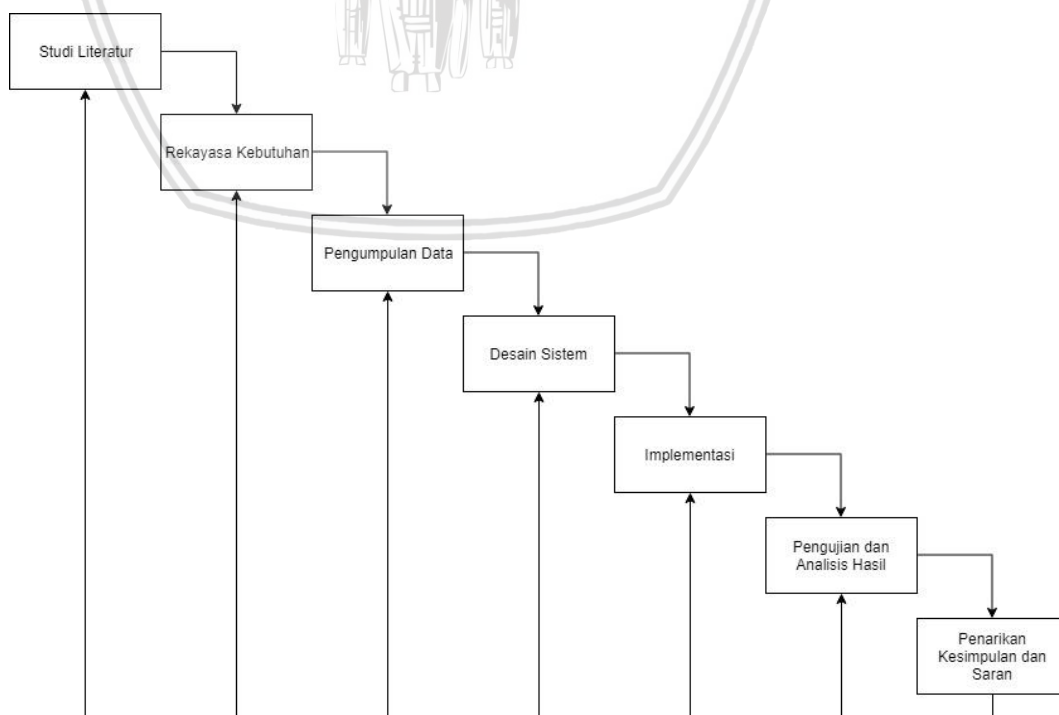




BAB 3 METODE PENELITIAN

3.1 METODE PENELITIAN

Bab ini menjelaskan metode yang digunakan dalam melakukan penelitian tentang perancangan *scoring system* otomatis pada lomba menembak dengan target *Sillhouette* hewan menggunakan metode klasifikasi *Naïve Bayes*. Tipe penelitian yang dilakukan adalah implementatif. Implementatif yaitu bersifat observasi dengan menggunakan rancang bangun yang sederhana. Langkah pertama yang dilakukan adalah mengumpulkan teori-teori pendukung penelitian dan dikemas dalam studi literatur. Kemudian dilanjutkan dengan proses analisis kebutuhan. Setelah tahap analisis kebutuhan, proses selanjutnya adalah melakukan proses perancangan yang dilanjutkan dengan proses implementasi *hardware* serta implementasi *software* sesuai dengan perancangan. Tahap berikutnya dilakukan pengujian dan analisis pada rancangan yang telah dibangun. Kesimpulan dan saran disertakan sebagai catatan atas rancangan dan kemungkinan arah pengembangan selanjutnya. Langkah-langkah yang dilakukan dalam penelitian ini ditunjukkan pada Gambar 3.1.



Gambar 3.1 Waterfall Diagram Metode Penelitian

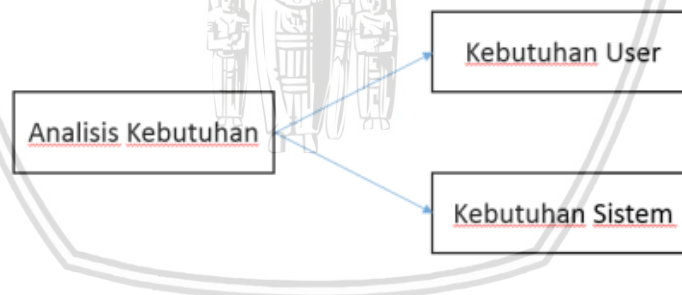
3.1.1 Studi Literatur

Studi Literatur adalah tahap awal yang digunakan untuk menambah studi pustaka dan pengetahuan yang mendukung untuk mengerjakan penulisan laporan dan penelitian. Studi literatur dilaksanakan dengan cara mengumpulkan teori dan pustaka yang berkaitan dengan penelitian ini meliputi:

1. Mekanisme hubungan sensor dengan mikrokontroler.
2. Pemrograman Arduino.
3. Mikrokontroler Arduino Mega 2560.

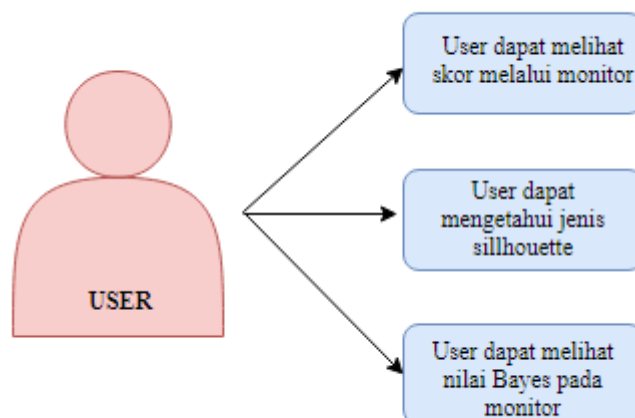
3.1.2 Analisis Kebutuhan

Pada tahap analisis kebutuhan dibagi menjadi dua bagian, yaitu analisis kebutuhan *user* dan analisis kebutuhan sistem sesuai dengan Gambar 3.2.



Gambar 3.2 Analisis Kebutuhan

Pada bagian analisis kebutuhan *user* didalamnya hanya menjelaskan kebutuhan secara fungsional. Terdapat beberapa poin yang menjelaskan hal yang dapat *user* lakukan pada sistem yang menjadi kebutuhan *user* terhadap sistem yang dibangun. Hal tersebut berkaitan dengan fitur yang terdapat pada system. Kebutuhan *user* dapat di definisikan seperti Gambae 3.3.



Gambar 3.3 Analisis Kebutuhan User

Pada tahap analisis kebutuhan sistem, terdapat kebutuhan secara fungsional dan non fungsional. Kebutuhan fungsional pada sistem ini menjelaskan tentang hal yang dibutuhkan pada sistem agar sistem yang dibangun dapat berjalan dengan baik. Sedangkan kebutuhan non fungsional pada sistem ini menjelaskan tentang hal yang menjadi batasan terhadap kebutuhan perancangan sistem. Adapun kebutuhan sistem dapat didefinisikan sebagai berikut:

Kebutuhan Non-Fungsional:

Perangkat Keras:

1. Mikrokontroler Arduino Mega 2560
2. Sensor *ADXL335 Accelerometer*
3. *Load Cell*
4. *Modul HX711*
5. *Limit Switch*
6. Power 5V
7. PC/Laptop

Perangkat Lunak:

1. Arduino IDE

Kebutuhan Fungsional:

Umum:

1. Jenis *Sillhouette* hewan yang digunakan
2. Jarak tembakan sejauh 5 meter

Khusus:

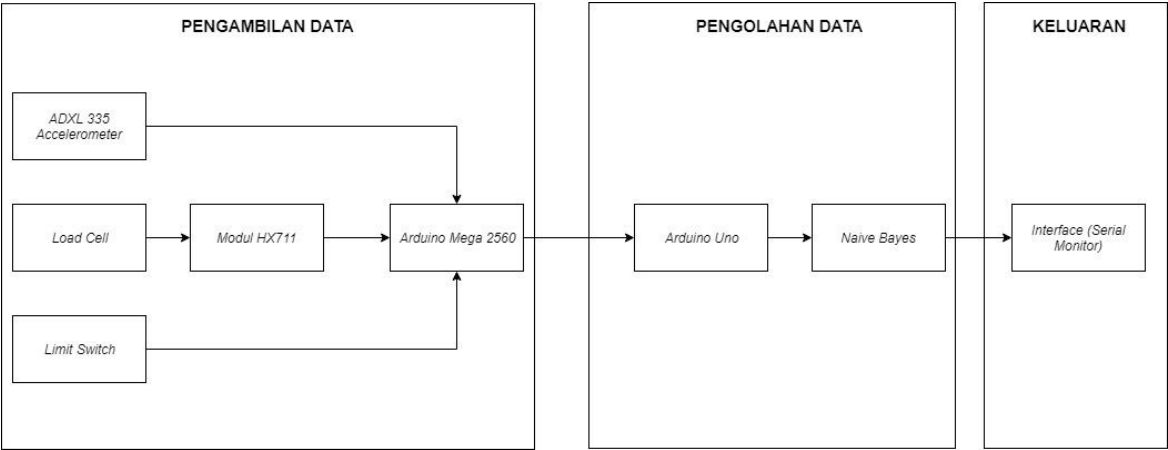
1. Sistem dapat mengetahui gerakan berdasarkan akselerasi pada *Sillhouette* dan berat dari *Sillhouette*.
2. Sistem dapat melakukan pengolahan *Naïve Bayes* berdasarkan *input* dari sensor gyro dan *Load Cell*.
3. Sistem dapat mengeluarkan *output* berupa nilai skor berdasarkan komputasi *Naïve Bayes*.

3.1.3 Pengambilan Data

Pengambilan data dengan cara mengambil nilai *output* dari tiga sensor yang berbeda yang masing-masing akan dijadikan acuan pengambilan keputusan pada klasifikasi *Naïve Bayes*.

3.1.4 Desain Sistem

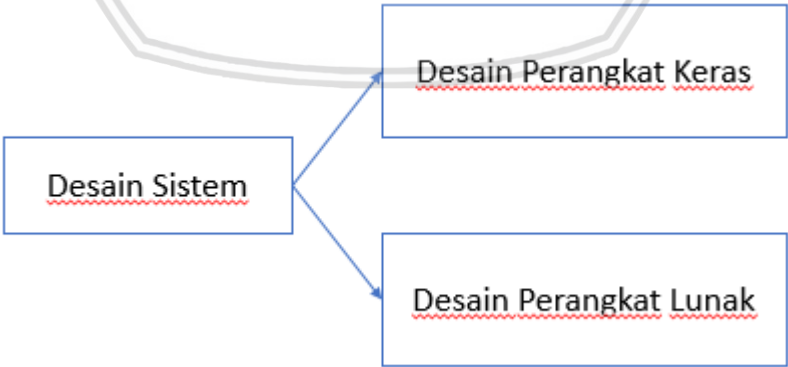
Desain sistem pada penelitian ini mendefinisikan sistem apa yang akan peneliti bangun. Adapun diagram blok sistem yang akan dibangun adalah seperti pada Gambar 3.4.



Gambar 3.4 Diagram Blok Rancangan Sistem

Pada diagram blok rancangan sistem, sistem yang akan dibangun berupa *prototype scoring system* otomatis pada lomba menembak dengan target *Sillhouette* hewan yang terhubung dengan sistem perangkat keras. Sistem perangkat keras saling terhubung dengan aplikasi perangkat lunak yaitu sebagai program pada sistem. Pada aplikasi perangkat lunak, terdapat program yang dapat menentukan nilai pada lomba menembak yang kemudian dapat menampilkan *outputnya* ke monitor.

Pada pembangunan keseluruhan sistem, diperlukan desain sistem yang lebih detail. Desain sistem terbagi menjadi dua bagian yaitu Desain Perangkat Keras dan Desain Perangkat Lunak seperti yang tercantum pada Gambar 3.5.



Gambar 3.5 Desain Sistem

Pada sistem desain akan mendeskripsikan desain sistem secara rancang *hardware*, rancang elektrik, dan *prototype*. Sedangkan pada

software desain akan menjelaskan desain program dan sistem klasifikasi *Naïve Bayes*.

3.1.5 Implementasi

Tahap implementasi sistem akan dilakukan berdasarkan perancangan yang telah ditentukan sebelumnya. Pada tahap ini akan diimplementasikan seluruh gagasan serta ide baik desain maupun secara perhitungan sistem yang telah menjadi target tujuan sebelumnya. Terdapat beberapa tahap implementasi yaitu :

1. Implementasi *prototype scoring system* otomatis
2. Implementasi Skematik sistem *hardware*
3. Implementasi metode klasifikasi *Naïve Bayes*

3.1.6 Pengujian dan Analisis Hasil

Pengujian dan analisis sistem dilakukan untuk mengetahui apakah kinerja dan performa keseluruhan sistem yang telah dirancang sesuai dengan spesifikasi kebutuhan yang melandasinya. Pengujian hanya dilakukan pada *mode* kontrol dengan beberapa skenario percobaan :

1. Target terlebih dahulu diberi gangguan berupa gerakan yg tidak stabil.
2. Pengujian *output* sistem dengan *metode klasifikasi Naïve Bayes*.

Sedangkan untuk parameter yang dianalisis adalah sebagai berikut :

1. Perbandingan *output* menggunakan *metode klasifikasi Naïve Bayes* yang sama.
2. Hasil skor berdasarkan *output* dari *metode klasifikasi Naïve Bayes*.

3.1.7 Penarikan Kesimpulan dan Saran

Kesimpulan didapatkan setelah melakukan tahap perancangan, implementasi, pengujian dan analisis terhadap sistem yang dibuat. Kesimpulan disusun berdasarkan hasil dari tahap pengujian dan analisis yang dilakukan pada sistem yang dibuat. Isi pada kesimpulan diharapkan dapat menjadi acuan dasar pada penelitian selanjutnya untuk mengembangkan *scoring system* otomatis pada lomba menembak. Pada akhir penulisan

terdapat saran yang bertujuan untuk memberikan kemudahan penelitian selanjutnya, apabila akan meneruskan dan mengembangkan penelitian ini.





BAB 4 REKAYASA KEBUTUHAN

Pada bab ini akan dijelaskan secara rinci terkait gambaran umum sistem, analisis kebutuhan fungsional dan non-fungsional sistem, kebutuhan perangkat keras dan perangkat.

4.1 Gambaran Umum Sistem

Scoring otomatis pada perlombaan menembak dengan target *Sillhouette* hewan dengan metode *Naïve Bayes* merupakan sebuah sistem yang dapat menentukan apakah target tersebut masuk kedalam kriteria *scoring* atau bukan. Sistem membutuhkan keadaan awal yang dimana belum diletakkannya target *Sillhouette* di atas tempat yang disediakan. Kemudian data sensor dari kontroler *input* diambil oleh sistem yang kemudian dikirimkan ke pusat kontroler. Kontroler akan mengambil data pada sensor *ADXL335*, *Load Cell*, dan *Limit Switch* yang kemudian dikirimkan ke pusat kontroler. *Naïve Bayes* digunakan untuk menentukan target tembak dan skor yang akan ditampilkan.

4.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan agar keseluruhan kebutuhan yang diperlukan untuk membangun *scoring* otomatis pada perlombaan menembak dengan target *Sillhouette* hewan dengan metode *Naïve Bayes* ini dapat dipenuhi. Dalam melakukan analisis kebutuhan sistem ini, akan dibagi menjadi 2 bagian yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

4.2.1 Kebutuhan Fungsional

Dibawah ini merupakan beberapa kebutuhan fungsional dari sistem yang harus dipenuhi, yaitu :

1. Sistem dapat mengetahui gerakan berdasarkan akselerasi pada *Sillhouette* dengan menggunakan sensor *ADXL 335 Accelerometer*.

Fungsi ini bertujuan untuk mengklasifikasi *Sillhouette* mana yang terjatuh. Untuk dapat mengetahui *Sillhouette* mana yang terjatuh, sistem akan berpatokan pada nilai dari sensor *ADXL335 Accelerometer*. Seperti yang diketahui bahwa sensor *ADXL335 Accelerometer* dapat mendeteksi

pergerakan suatu benda berdasarkan akselerasinya. Hasil dari sensor ini akan dijadikan *input* pada sistem. Nilai *input* dari sistem ini akan dikomputasikan oleh kontroler pusat untuk melakukan klasifikasi *Naïve Bayes*.

2. Sistem dapat mengetahui berat dari *Sillhouette* dengan menggunakan sensor *Load Cell*.

Fungsi ini bertujuan untuk mengklasifikasi jenis *Sillhouette* yang ada pada sistem. Pada dasarnya *Sillhouette* yang digunakan penulis pada sistem ada 4 buah dengan berat yang berbeda. *Sillhouette* ayam menjadi yang teringan pada sistem ini, dan *Sillhouette* domba menjadi yang terberat pada sistem ini. Dikarenakan berat dari *Sillhouette* yang berbeda, penulis dapat mengklasifikasikan jenis *Sillhouette* berdasarkan berat. Dalam pengklasifikasian berdasarkan berat ini menggunakan sensor *Load Cell*. Nilai *input* dari sistem ini akan dikomputasikan oleh kontroler pusat untuk melakukan klasifikasi *Naïve Bayes*.

3. Sistem dapat melakukan pengolahan *Naïve Bayes* berdasarkan *input* dari sensor *ADXL335* dan *Load Cell*.

Fungsi ini berperan penting agar proses klasifikasi *Naïve Bayes* dapat menghasilkan *output* yang sesuai. Apabila fungsi ini tidak berjalan maka sistem tidak akan menghasilkan *output* yang akurat sesuai dengan perhitungan klasifikasi *Naïve Bayes*. Nilai sensor diambil satu-satu per kotak *Sillhouette* yang kemudian dimasukan kedalam komputasi *Naïve Bayes*.

4. Sistem dapat mengeluarkan *output* berupa nilai skor berdasarkan komputasi *Naïve Bayes*.

Fungsi ini mengharapkan saat seluruh proses klasifikasi *Naïve Bayes* telah dilakukan dan sistem dapat menentukan *output*, maka monitor akan mengeluarkan nilai skor.

5. *Sillhouette* ditembak berurutan mulai dari yang paling kiri hingga yang paling kanan.

Fungsi ini mengacu pada aturan yang telah dibuat perbakin untuk lomba menembak Metal *Silhouette*. Aturannya berbunyi “Sasaran ditembak dari kiri kekanan, apabila meleset atau tidak kena tembakan diteruskan ke sasaran berikutnya, sampai jumlah tembakan dan waktu habis” (JEHACESHOOTINGCLUB, 2017). Jika waktu penembakan tiap *silhouette* habis, maka penembak diharuskan berpindah sasaran. Penembak akan di diskualifikasi ketika menembak *silhouette* yang tidak dianjurkan dan skor tidak dihitung.

6. Sistem memiliki 4 mode penembakan.

Fungsi ini bertujuan memberikan 4 mode penembakan yang berbeda pada sistem ini. 4 mode yang dimaksud adalah mode ayam, mode kalkun, mode babi dan mode domba.

4.2.2 Kebutuhan Non-Fungsional

Ada 2 kebutuhan non-fungsional pada sistem ini, yaitu kebutuhan perangkat keras dan kebutuhan perangkat lunak.

4.2.2.1 Kebutuhan Perangkat Keras

Berikut ini merupakan beberapa perangkat keras yang dibutuhkan untuk membuat *prototype* dari sistem, yakni :

1. Mikrokontroler Arduino Uno

Mikrokontroler *Arduino Uno* digunakan sebagai pusat kontroler yang berfungsi sebagai inti dari pengolahan data dari *prototype* sistem yang dibuat.

2. Mikrokontroler Arduino Mega 2560

Mikrokontroler *Arduino Mega 2560* digunakan sebagai kontroler *input* yang menampung data yang diambil sensor yang nantinya akan dikirimkan ke pusat kontroler.

3. Sensor ADXL335

Sensor *ADXL335* atau biasa disebut sensor GY-61 *Accelerometer* digunakan untuk mengukur gerakan ketika *Silhouette* jatuh. *Output* dari sensor ini akan digunakan sebagai *input* dari proses *Naïve Bayes*.

4. *Load Cell*

Loadcell merupakan sensor berat yang memiliki fungsi sebagai mengukur berat dari *Sillhouette* yang terpasang pada target.

5. *Modul HX711*

Modul HX711 digunakan sebagai perantara *Load Cell* dengan mikrokontroler.

6. *Limit Switch*

Limit Switch digunakan sebagai pemberi *trigger* untuk melakukan pengiriman data dari kontroler input ke pusat kontroler.

7. Laptop

Pada penelitian ini laptop digunakan untuk membuat program pada sistem ini baik program untuk Arduino Mega 2560, *Arduino Uno* maupun Delphi7. Laptop juga digunakan sebagai monitor untuk menampilkan skor.

4.2.2.2 Kebutuhan Perangkat Lunak

Sedangkan untuk kebutuhan perangkat lunak terdapat beberapa yang dibutuhkan dalam *prototype* dari sistem ini, yaitu :

1. Library wire.h

Library wire.h digunakan sebagai jembatan komunikasi antar kontroler menggunakan I2C.

2. Library HX711.h

Library HX711.h digunakan sebagai penunjang kerja serta komunikasi antara sensor *Load Cell* dengan mikrokontroler.

3. *Naïve Bayes*

Metode *Naïve Bayes* digunakan sebagai penentu keputusan *output* yang akan dikeluarkan oleh sistem.

4.3 Batasan Desain Sistem

Dalam pembuatan *prototype* dari *scoring* otomatis pada perlombaan menembak dengan target *Sillhouette* hewan dengan metode *Naïve Bayes* terdapat Batasan yang diterapkan sehingga dalam proses perancangan, implementasi serta

pembahasan dari *prototype* sistem ini tidak mencakup jangkauan yang luas. Batasan-batasan dari desain *prototype* sistem tersebut yaitu :

1. Sistem memperoleh berat awal dari *input Load Cell* ketika *Sillhouette* belum diletakkan pada target.
2. Sistem melakukan perhitungan jumlah *Sillhouette* yang dijadikan acuan *input* prototipe sistem.
3. Sistem melakukan konfirmasi apakah *Sillhouette* yang jatuh merupakan yang ditargetkan atau bukan.
4. Data latih *Load Cell* diambil ketika *Sillhouette* diletakan diatas *box*.
5. Data latih sensor *ADXL 335 Accelerometer* diambil ketika *Sillhouette* dalam posisi normal atau tidak tertembak dan *Sillhouette* dalam keadaan tertembak.





BAB 5 PERANCANGAN DAN IMPLEMENTASI

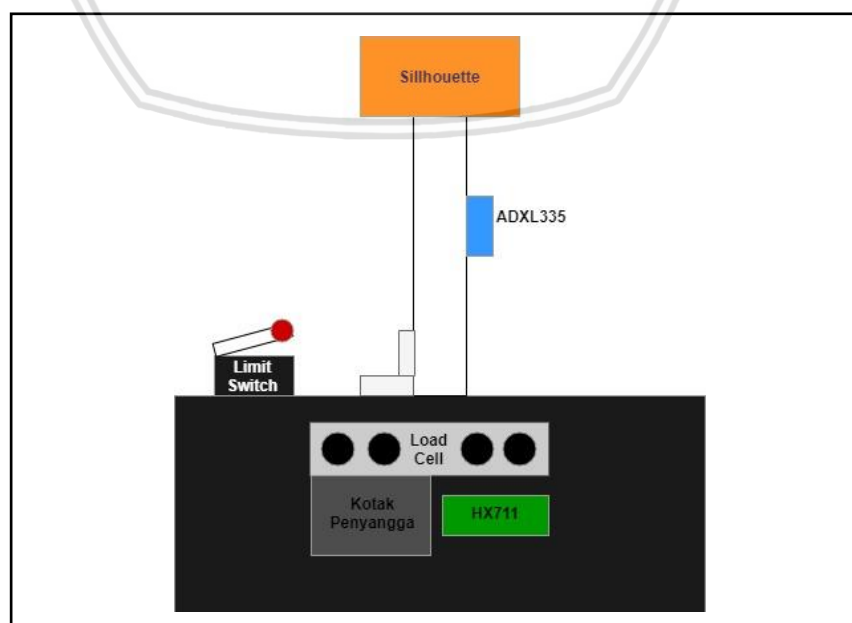
Pada bab ini akan menjelaskan proses perancangan dan implementasi dari sistem. Terdapat dua perancangan yaitu perancangan perangkat keras (*Hardware*) dan perancangan perangkat lunak (*Software*). Sedangkan pada implementasi perangkat keras (*Hardware*) dan implementasi terdapat implementasi perangkat lunak (*Software*).

5.1 Perancangan Sistem

Pada perangkat yang dirancang, sistem dapat melakukan *scoring* berdasarkan dari berat pada *Sillhouette*, dan orientasi gerakan pada *Sillhouette* ketika ditembak. *Input* berupa berat dan getaran pada *Sillhouette* yang diolah menggunakan mikrokontroler. *Output* pada sistem menggunakan metode klasifikasi *Naïve Bayes* untuk menentukan apakah *Sillhouette* yang jatuh termasuk dalam mode penembakan.

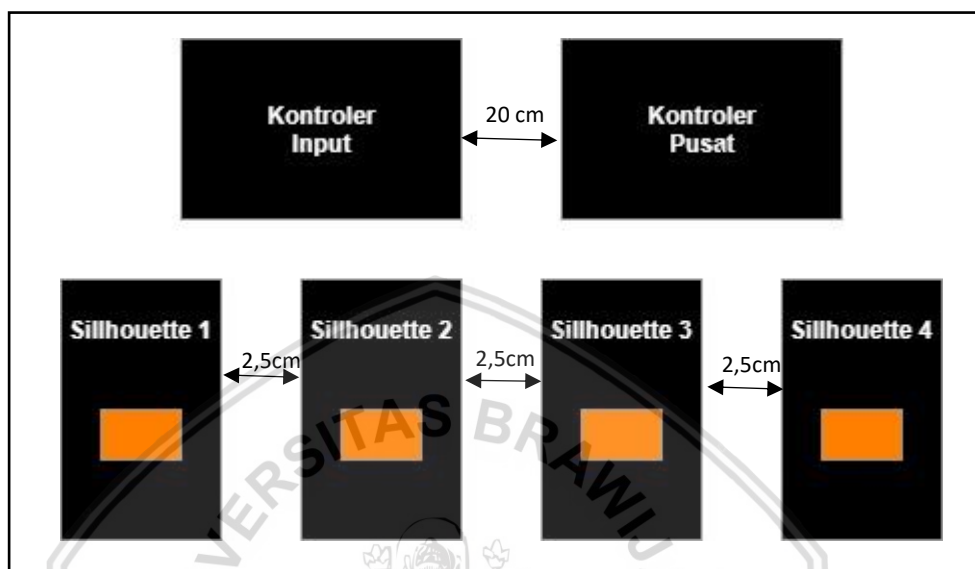
5.1.1 Perancangan *Prototype Scoring System*

Dalam perancangan *Prototype Scoring System* menggunakan *Load Cell*, sensor *ADXL335*, *Modul HX711*, dan *Limit Switch* yang diletakan pada kotak penyimpanan berbahan plastik.



Gambar 5.1 *Prototype* Alat Tampak Samping

Pada Gambar 5.1 dijelaskan bahwa bagian atas kotak *Sillhouette* terdapat *Limit Switch*, sensor *ADXL335* serta *Sillhouettenya*, sedangkan bagian dalam kotak, terdapat *Load Cell*, Modul *HX711*.

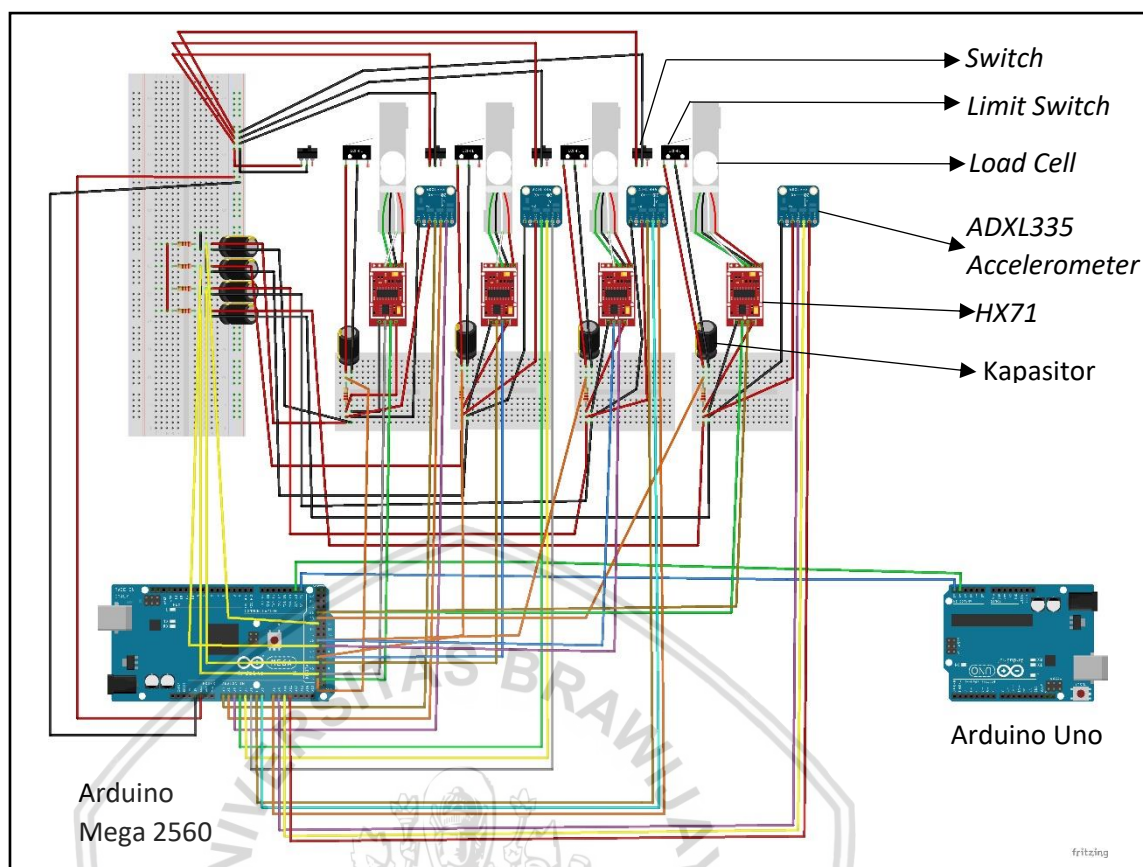


Gambar 5.2 *Prototype* Alat Tampak Atas

Pada Gambar 5.2 dijelaskan bahwa jarak antara kontroler input dan pusat kontroler berjarak 6cm, sedangkan jarak antara silhouette 1, silhouette 2, silhouette 3, dan silhouette 4 berjarak 2,5cm. Gambar 5.2 merupakan penampang alat tampak atas.

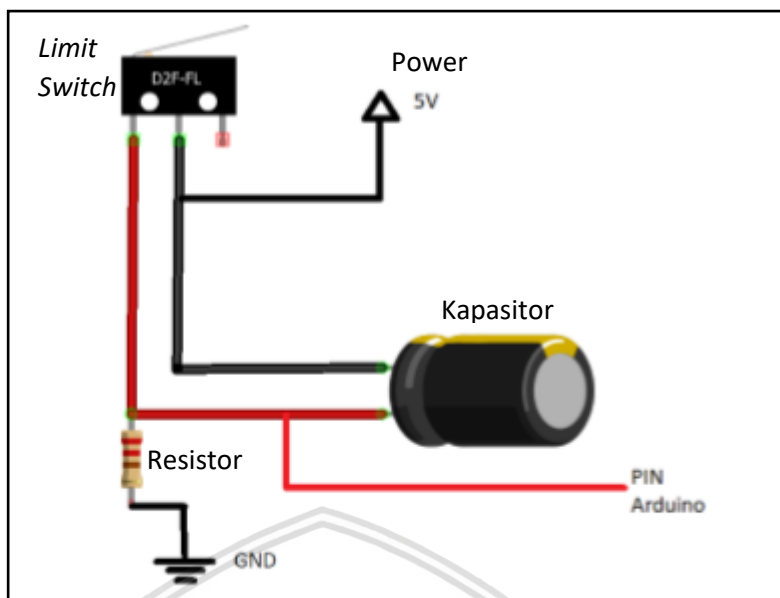
5.1.2 Perancangan Perangkat Keras

Perangkat keras (*Hardware*) digunakan untuk membangun *scoring system* yang mendukung mikrokontroler untuk menerapkan klasifikasi *Naïve Bayes* sebagai pengambil keputusan *output*. Mikrokontroler yang digunakan adalah Arduino Mega 2560. *Input* yang digunakan untuk mengukur berat *Sillhouette* adalah *Load Cell* dan sensor *ADXL335* untuk mengukur pergerakan posisi pada *Sillhouette*.



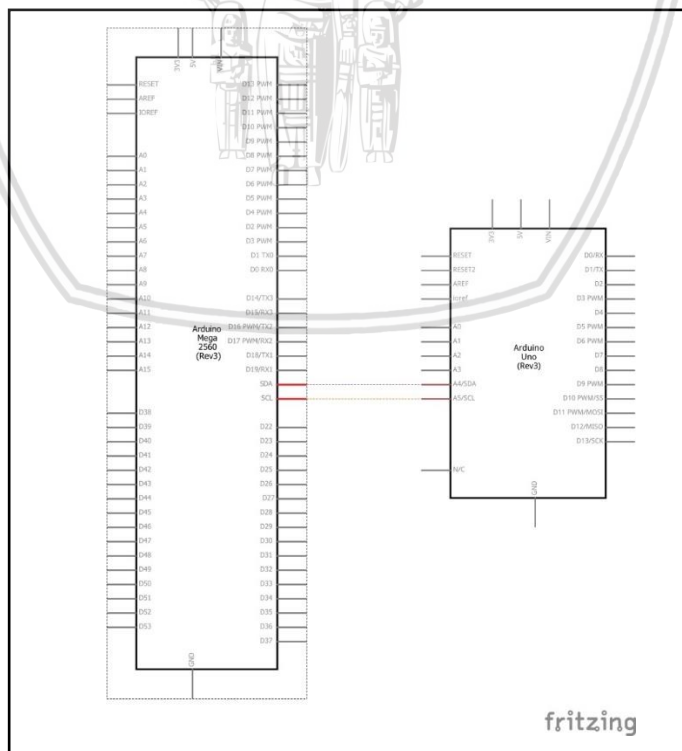
Gambar 5.3 Skema Perancangan Perangkat Keras

Gambar 5.3 merupakan skema dari perancangan perangkat keras, *input* data berupa sensor *ADXL335*, *Load Cell*, dan *Limit Switch*. Selanjutnya data pada sensor *ADXL335*, *Load Cell*, dan *Limit Switch* yang telah diambil akan di proses pada Arduino sehingga sistem mengetahui skor dan jenis *Sillhouette* yang tertembak. Kemudian hasil dari proses yang dilakukan pada Arduino akan ditampilkan melalui monitor. Pada *Limit Switch* terdapat rangkaian *Debounce* untuk menghilangkan efek *bouncing* yang terjadi pada *Limit Switch*, karena efek *bouncing* ini akan menyebabkan nilai dari variabel *Limit Switch* tidak akurat, ketika ditekan *Limit Switch* value nya tidak tentu antara 0 atau 1. Rangkaian *Debounce* yang dihubungkan pada *Limit Switch* terdapat pada Gambar 5.4.



Gambar 5.4 Rangkaian Debounce

Untuk komunikasi antar *Arduino Uno* dan *Arduino Mega* menggunakan *I2C*. *Arduino Mega* digunakan sebagai master writer dan *Arduino Uno* sebagai slave receiver. Koneksi *I2C* antara *Arduino Uno* dan *Arduino Mega* terdapat pada Gambar 5.5.



Gambar 5.5 Koneksi I2C antar Arduino

Koneksi pin Gambar 5.5 ditunjukkan pada Tabel 5.1.

Tabel 5.1 Koneksi pin perangkat keras

Pin Arduino Mega 2560	Pin Arduino Uno	Pin ADXL 335 (Accelerometer)	Pin Load Cell	Pin Limit Switch	Pin Switch (State Push)
Vcc	-	Vcc	Vcc	Vcc	Vcc
GND	GND	GND	GND	GND	GND
SDA	A5	-	-	-	-
SCL	A4	-	-	-	-
A0	-	x1	-	-	-
A1	-	y1	-	-	-
A2	-	z1	-	-	-
A3	-	x2	-	-	-
A4	-	y2	-	-	-
A5	-	z2	-	-	-
A6	-	x3	-	-	-
A7	-	y3	-	-	-
A8	-	z3	-	-	-
A9	-	x4	-	-	-
A10	-	y4	-	-	-
A11	-	z4	-	-	-
22	-	-	-	-	State Push 1
23	-	-	-	Limit 1	-
24	-	-	SCKOUT1	-	-
25	-	-	DOUT1	-	-
30	-	-	-	-	State Push 2
31	-	-	-	Limit 2	-
32	-	-	SCK2	-	-
33	-	-	DOUT2	-	-

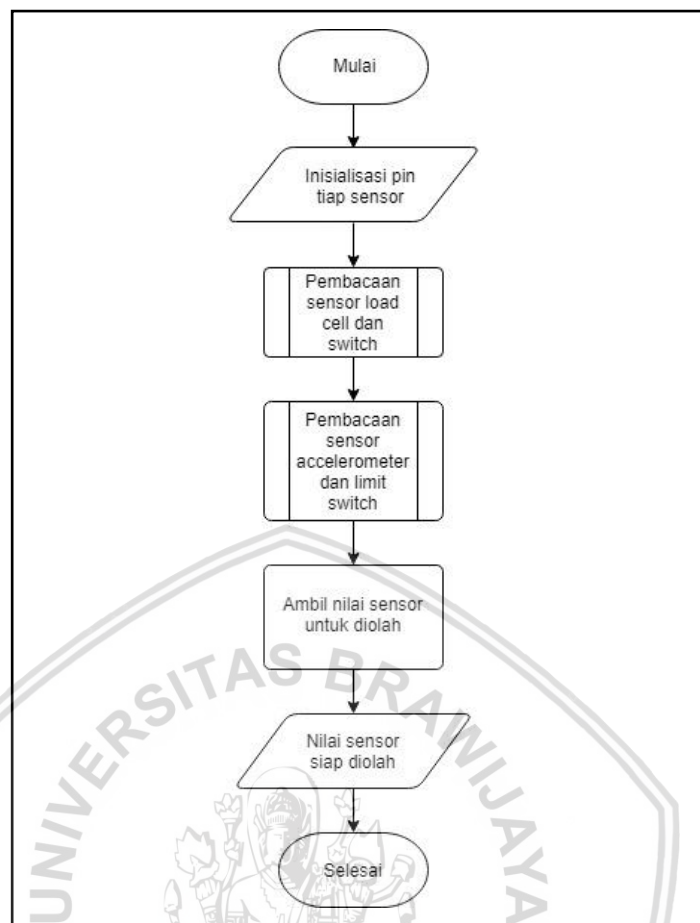
Pin Arduino Mega 2560	Pin Arduino Uno	Pin ADXL 335 (Accelerometer)	Pin Load Cell	Pin Limit Switch	Pin Switch (State Push)
38	-	-	-	-	State Push 3
39	-	-	-	Limit 3	-
40	-	-	SCK3	-	-
41	-	-	DOUT3	-	-
46	-	-	-	-	State Push 4
47	-	-	-	Limit 4	-
48	-	-	SCK4	-	-
49	-	-	DOUT4	-	-

5.1.3 Perancangan Perangkat Lunak

Pada sub bab perancangan perangkat lunak dibagi menjadi 3 pembahasan, yakni perancangan perangkat lunak untuk pembacaan data sensor, melakukan proses klasifikasi dengan metode *Naive Bayes* dan perancangan perangkat lunak untuk *scoreboard*.

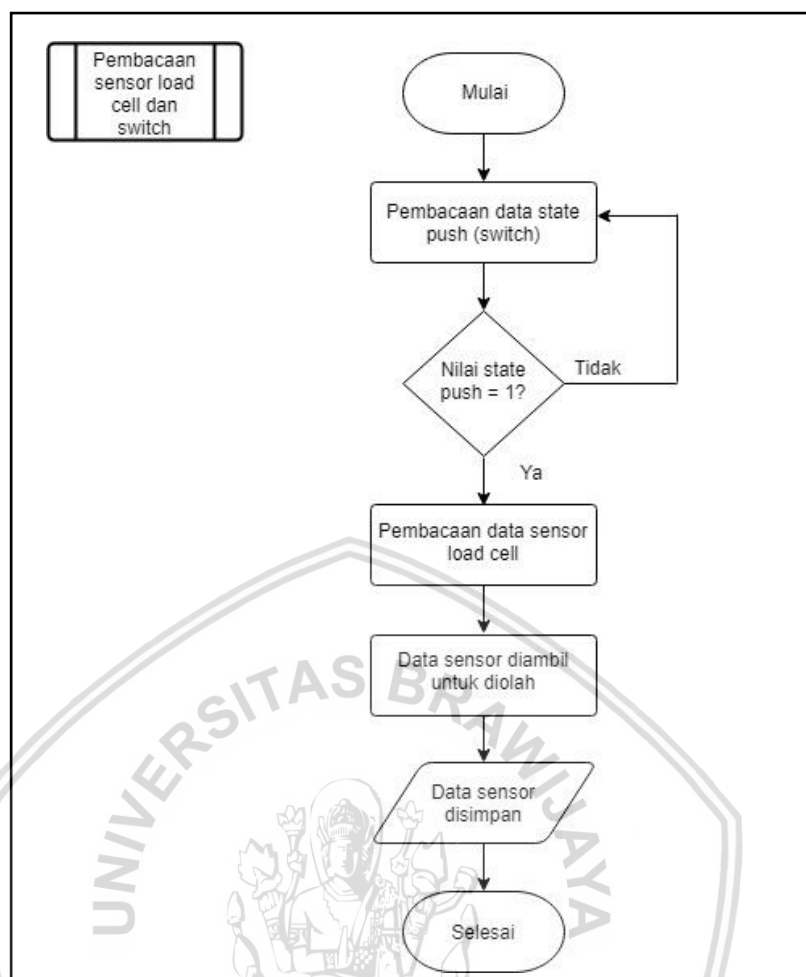
5.1.3.1 Perancangan Perangkat Lunak Pembacaan Data Sensor

Pada perancangan perangkat lunak (*Software*) untuk pengambilan data sensor dimaksudkan untuk menentukan *output* pembacaan data sensor mana yang akan diolah untuk dilakukan klasifikasi dengan menggunakan *Naive Bayes*, karena pada dasarnya mikrokontroler akan melakukan pembacaan nilai sensor secara terus menerus. *Flowchart* dari perancangan perangkat lunak (*Software*) pembacaan data sensor terdapat pada Gambar 5.6.



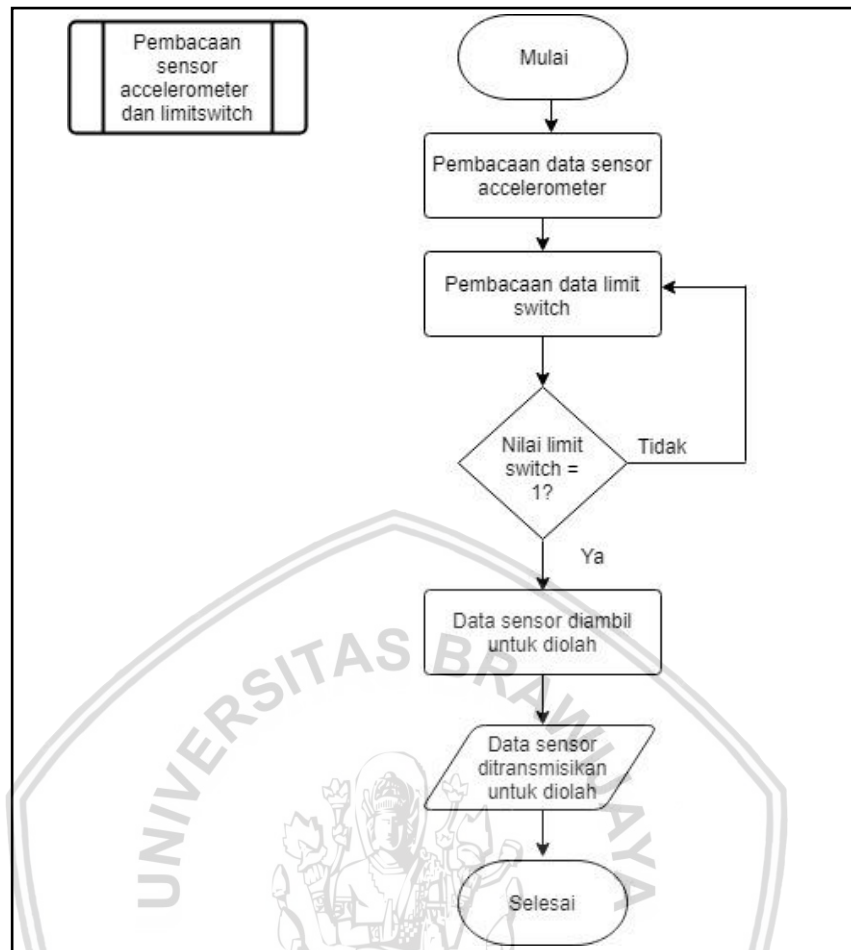
Gambar 5.6 Flowchart Perancangan Software pembacaan sensor

. Pada sistem ini dimulai dengan inisialisasi pin tiap sensor. Setelah itu masuk ke pembacaan sensor *Load Cell* dan *Switch*. Disini terdapat proses lain yang akan dijelaskan pada Gambar 5.7. Setelah pembacaan sensor *Load Cell* dan *Switch*, sistem masuk ke pembacaan sensor *Accelerometer* dan *Limit Switch*. Seperti sebelumnya, pada proses ini terdapat Sub Proses atau proses tambahan yang akan dijelaskan pada Gambar 5.8. Setelah itu data sensor diambil sehingga data siap diolah.



Gambar 5.7 Flowchart Perancangan Software pembacaan sensor *Load Cell & Switch*

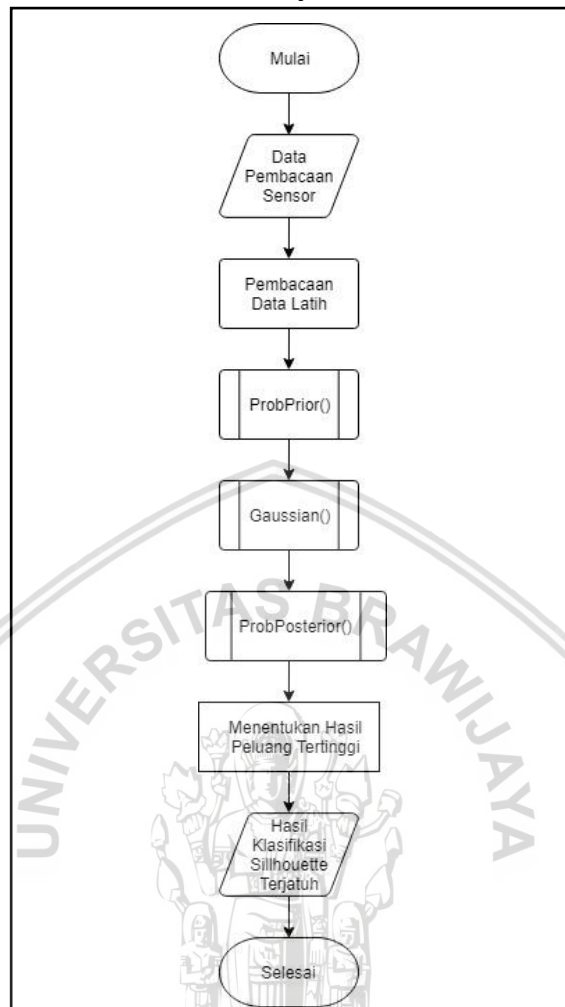
Pada Gambar 5.7, pembacaan sensor *Load Cell* dan *Switch* dimulai dari pembacaan data state push (*Switch*) terlebih dahulu. Kemudian sistem akan memberi konfirmasi apakah nilai state push (*Switch*) = 1 atau tidak. Jika tidak maka akan kembali ke proses sebelumnya, namun jika iya maka sistem akan melakukan pembacaan sensor *Load Cell*. Jika data sensor *Load Cell* sudah dibaca, maka data tersebut akan diambil. Kemudian data yang sudah diambil akan disimpan sebagai salah satu parameter klasifikasi *Sillhouette*.



Gambar 5.8 Flowchart Perancangan Software pembacaan sensor Accelerometer & Limit Switch

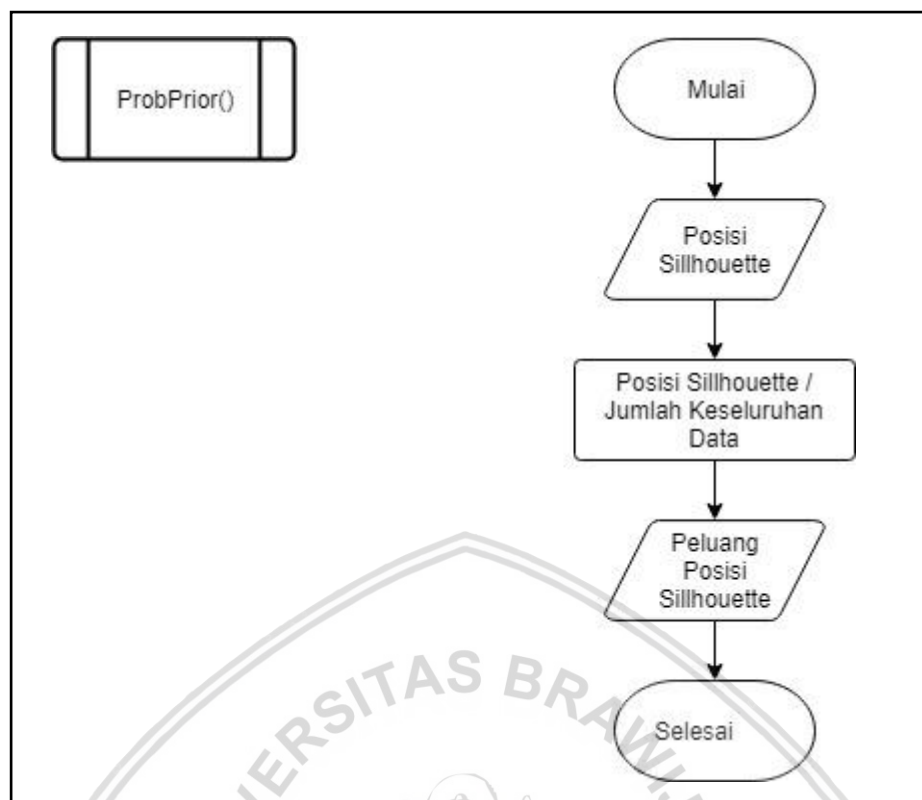
Pada Gambar 5.8, pembacaan *Accelerometer* dan *Limit Switch* dimulai dari pembacaan data sensor *Accelerometer* terlebih dahulu. Setelah itu sistem akan membaca data sensor *Limit Switch*. Lalu kemudian sistem akan memberi konfirmasi apakah data sensor *Limit Switch* = 1 atau tidak. Jika tidak maka akan kembali ke proses sebelumnya, namun jika iya maka data sensor *Load Cell* dan *Accelerometer* akan di kirimkan ke *Arduino Uno* untuk melakukan proses klasifikasi *Naïve Bayes*.

5.1.3.2 Perancangan Klasifikasi *Naïve Bayes*



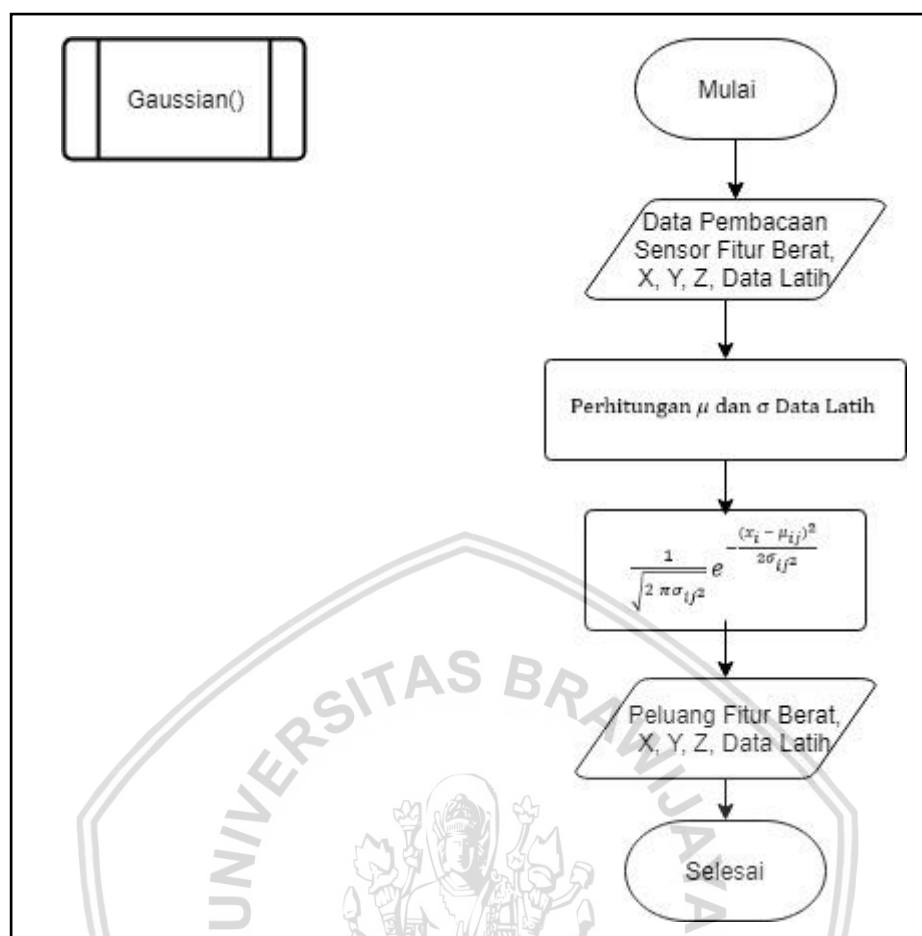
Gambar 5.9 Flowchart Perancangan Klasifikasi *Naïve Bayes*

Dalam melakukan proses klasifikasi menggunakan metode *Naive Bayes* terdapat beberapa tahap sesuai dengan Gambar 5.9 dimana yang menjadi masukan awal adalah nilai dari hasil pembacaan sensor. Hasil pembacaan sensor ini lah yang akan menjadi fitur-fitur yang mempengaruhi penentuan klasifikasi jenis silhouette mana yang terjatuh, selain itu hasil klasifikasi juga dipengaruhi oleh nilai dari data latih. Proses dimulai dari menentukan hasil dari fungsi *ProbPrior()*, menentukan hasil dari dari fungsi *Gaussian()*, menentukan hasil dari fungsi *ProbPosterior()*, menentukan hasil peluang tertinggi hingga didapatkan hasilnya. Adapun penjelasan dari masing-masing fungsi ditunjukkan dan dibahas sesuai dengan beberapa *Flowchart*.



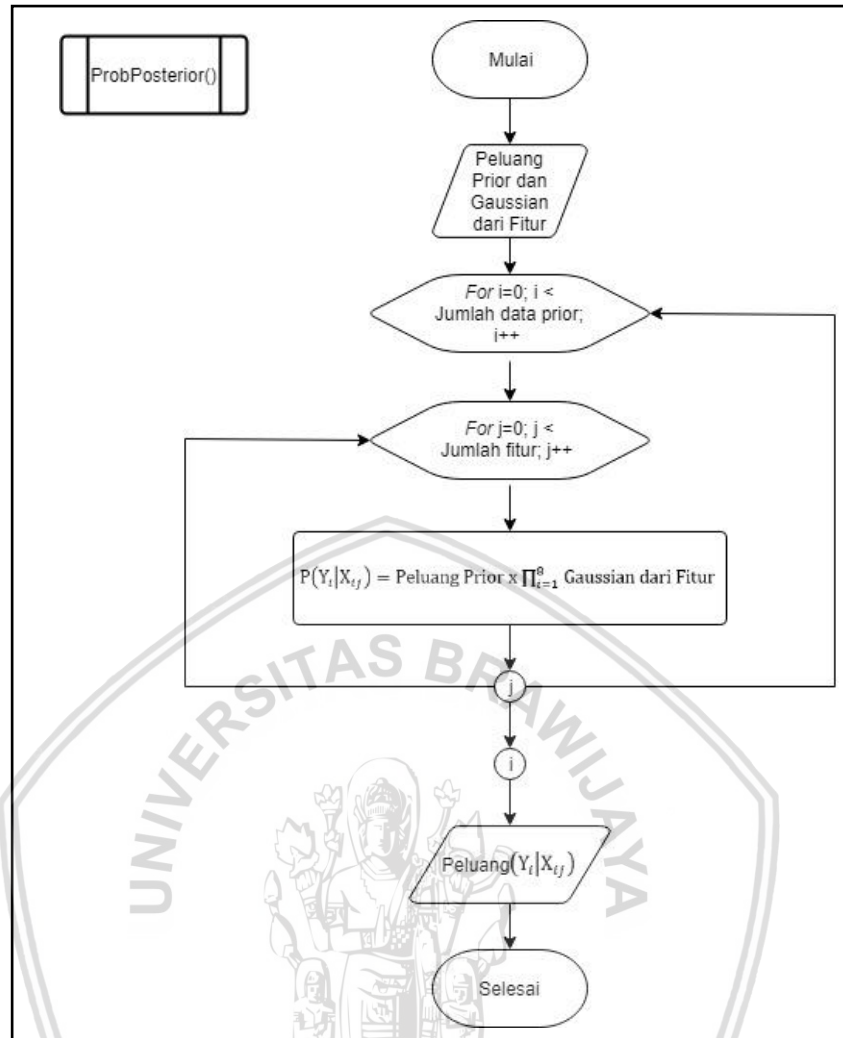
Gambar 5.10 Flowchart Fungsi *ProbPrior()*

Tahap pertama yang dilakukan dalam mengklasifikasikan posisi *Sillhouette* dengan metode Naive Bayes adalah menghitung nilai prior dari masing kelas. Seperti yang dijelaskan pada Gambar 5.10, nilai prior merupakan nilai peluang terjadinya suatu kelas dengan cara membagi banyaknya data dalam suatu kelas (dalam sistem ini terdapat 8 kelas yaitu Ayam Tertembak, Ayam Tidak Tertembak, Kalkun Tertembak, Kalkun Tidak Tertembak, Babi Tertembak, Babi Tidak Tertembak, Domba Tertembak dan Domba Tidak Tertembak) dengan jumlah keseluruhan data yang ada. Data yang dilakukan perhitungan nilai prior adalah data latih.



Gambar 5.11 Flowchart Fungsi Gaussian()

Tahap kedua yaitu untuk menentukan nilai peluang dari masing-masing fitur. Terdapat 4 fitur yang digunakan dalam sistem ini yaitu 3 fitur dari pembacaan sensor *Accelerometer ADXL335* (fitur X, fitur Y dan fitur Z) serta 1 fitur lainnya dari pembacaan sensor *Load Cell* yaitu berat dari *Silhouette*. Namun sebelum dapat menentukan nilai peluang dari masing-masing fitur, terlebih dahulu harus dilakukan perhitungan mean dan standar deviasi dari data latih menggunakan Persamaan (2.5) dan Persamaan (2.6). Pada sistem ini, data latih yang berupa nilai mean dan standar deviasi disimpan pada pemrograman mikrokontroler untuk mempermudah dalam mengakses nilai dari data latih saat sistem dijalankan. Selanjut perhitungan Gaussian dilakukan dengan menggunakan Persamaan (2.4) seperti yang ditunjukkan pada Gambar 5.11.



Gambar 5.12 Flowchart Fungsi ProbPosterior()

Tahap selanjutnya adalah menentukan nilai dari peluang *posterior*, peluang *posterior* yaitu peluang untuk menentukan besarnya peluang masing-masing kelas akan terjadi ketika adanya masukan dari tiap fitur. Pada sistem ini yaitu untuk menentukan besarnya peluang masing-masing kondisi *Sillhouette* akan terjadi ketika adanya pembacaan nilai fitur X, fitur Y, fitur Z dan fitur Berat. Seperti penjelasan pada Gambar 5.12, proses perhitungan *ProbPosterior()* dengan melakukan perkalian antara hasil dari fungsi *ProbPrior()* dengan fungsi *Gaussian()*. Tahap akhir dalam pengklasifikasian dengan *Naïve Bayes* ini adalah menentukan nilai peluang *posterior* yang tertinggi dengan cara membandingkan satu sama lain antar peluang *posterior*. Kondisi *Sillhouette* dengan nilai peluang *posterior* paling tinggi merupakan hasil yang klasifikasi kondisi *Sillhouette* yang dideteksi oleh sistem. Contoh perhitungan manual untuk melakukan klasifikasi terhadap

silhouette kalkun dengan data uji yang mempunyai fitur $X=3,54$, $Y=0,68$, $Z=13,19$ dan Berat=36,21 gram berdasarkan data latih pada LAMPIRAN adalah sebagai berikut :

Sebelum mengawali perhitungan, alangkah lebih baik dibaca terlebih dahulu keterangan yang tertera pada tabel 5.2 di bawah ini :

Tabel 5.2 Keterangan dari singkatan masing-masing *Sillhouette*

Keterangan	
AT	Ayam Tertembak
ATT	Ayam Tidak Tertembak
KT	Kalkun Tertembak
KTT	Kalkun Tidak Tertembak
BT	Babi Tertembak
BTT	Babi Tidak Tertembak
DT	Domba Tertembak
DTT	Domba Tidak Tertembak

1. Menghitung probabilitas *Prior* dari masing-masing *Sillhouette*

$$P_{AT} : \frac{\text{Jumlah jenis Sillhouette AT}}{\text{Jumlah seluruh data}} = \frac{10}{80} = 0,125$$

$$P_{ATT} : \frac{\text{Jumlah jenis Sillhouette ATT}}{\text{Jumlah seluruh data}} = \frac{10}{80} = 0,125$$

$$P_{KT} : \frac{\text{Jumlah jenis Sillhouette KT}}{\text{Jumlah seluruh data}} = \frac{10}{80} = 0,125$$

$$P_{KTT} : \frac{\text{Jumlah jenis Sillhouette KTT}}{\text{Jumlah seluruh data}} = \frac{10}{80} = 0,125$$

$$P_{BT} : \frac{\text{Jumlah jenis Sillhouette BT}}{\text{Jumlah seluruh data}} = \frac{10}{80} = 0,125$$

$$P_{BTT} : \frac{\text{Jumlah jenis Sillhouette BTT}}{\text{Jumlah seluruh data}} = \frac{10}{80} = 0,125$$

$$P_{DT} : \frac{\text{Jumlah jenis Sillhouette DT}}{\text{Jumlah seluruh data}} = \frac{10}{80} = 0,125$$

$$P_{DTT} : \frac{\text{Jumlah jenis Sillhouette DTT}}{\text{Jumlah seluruh data}} = \frac{10}{80} = 0,125$$

1. Menghitung nilai *Mean* dan Standar Deviasi dari masing-masing *Sillhouette*

a. *Mean*

$$\begin{aligned}\bar{x}_{berat(KT)} &= \frac{\sum_{i=0}^n KT_i}{Jumlah\ jenis\ Sillhouette\ KT} \\ &= \frac{36,21+36,66+36,83+35,36+33,99+33,19+36,62+35,43+36,04+37,02}{10} \\ &= 35,735\end{aligned}$$

$$\begin{aligned}\bar{x}_{X(KT)} &= \frac{\sum_{i=0}^n KT_i}{Jumlah\ jenis\ Sillhouette\ KT} \\ &= \frac{5,2+4+22,69+3,84-21,48+1,58-26,76-6,26-21,79-11,84}{10} \\ &= -5,082\end{aligned}$$

$$\begin{aligned}\bar{x}_{Y(KT)} &= \frac{\sum_{i=0}^n KT_i}{Jumlah\ jenis\ Sillhouette\ KT} \\ &= \frac{8,67+8,37+7,61-3,39+2,49+0,68-3,38-0,83-0,38+0,23}{10} \\ &= 2,007\end{aligned}$$

$$\begin{aligned}\bar{x}_{Z(KT)} &= \frac{\sum_{i=0}^n KT_i}{Jumlah\ jenis\ Sillhouette\ KT} \\ &= \frac{21,33+22,69+6,61+23,14+8,97+10,93+6,71+21,03+1,13+24,8}{10} \\ &= 14,734\end{aligned}$$

Perhitungan yang sama dilakukan untuk mencari *mean* dari jenis *Sillhouette* KTT, ATT, AT, BTT, BT, DTT, dan DT. Nilai *mean* diambil dengan cara menjumlahkan semua nilai data latih sensor masing-masing kelas, lalu dibagi dengan jumlah data latih sensor masing-masing kelas. Hasil perhitungan *mean* keseluruhan jenis *Sillhouette* ditunjukkan pada Tabel 5.3. Pada Tabel 5.3 dibawah ini merepresentasikan nilai *mean* masing-masing fitur (data sensor) terhadap masing-masing kelas. *Mean* akan diinisialisasikan pada kode program sebagai global variabel, yang berguna untuk perhitungan *gaussian* dan *posterior* pada program.

Tabel 5.3 Mean masing-masing *Sillhouette*

	Berat	X	Y	Z
AT	28,12	-9,333	-3,302	18,859
ATT	29,102	13,6	1,375	4,63
KT	35,735	-5,082	2,007	14,734
KTT	35,858	13,46	2,4	5,185
BT	47,533	-2,594	18,198	0,305
BTT	48,235	13,377	2,142	2,58
DT	65,956	-5,079	7,25	14,97
DTT	68,064	13,555	2,475	2,58

b. Standar Deviasi

$$\sigma_{berat(KT)} = \sqrt{\frac{\sum_{i=1}^n (berat_i - \bar{x}_{berat(KT)})^2}{Jumlah\ jenis\ Sillhouette\ KT - 1}}$$

$$= \sqrt{\frac{(36,21 - 35,735)^2 + (36,66 - 35,735)^2 + (36,83 - 35,735)^2 + (35,36 - 35,735)^2 + (33,99 - 35,735)^2 + (33,19 - 35,735)^2 + (36,62 - 35,735)^2 + (35,43 - 35,735)^2 + (36,04 - 35,735)^2 + (37,02 - 35,735)^2}{10 - 1}}$$

$$= 1,27207$$

$$\sigma_{X(KT)} = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{x}_{X(KT)})^2}{Jumlah\ jenis\ Sillhouette\ KT - 1}}$$

$$= \sqrt{\frac{(5,2 + 5,082)^2 + (4 + 5,082)^2 + (22,69 + 5,082)^2 + (3,84 + 5,082)^2 + (-21,48 + 5,082)^2 + (1,58 + 5,082)^2 + (-26,76 + 5,082)^2 + (-6,26 + 5,082)^2 + (-21,79 + 5,082)^2 + (-11,84 + 5,082)^2}{10 - 1}}$$

$$= 15,4507$$

$$\sigma_{Y(KT)} = \sqrt{\frac{\sum_{i=1}^n (Y_i - \bar{x}_{Y(KT)})^2}{Jumlah\ jenis\ Sillhouette\ KT - 1}}$$

$$= \sqrt{\frac{(8,67 - 2,007)^2 + (8,37 - 2,007)^2 + (7,61 - 2,007)^2 + (-3,39 - 2,007)^2 + (2,49 - 2,007)^2 + (0,68 - 2,007)^2 + (-3,38 - 2,007)^2 + (-0,83 - 2,007)^2 + (-0,38 - 2,007)^2 + (0,23 - 2,007)^2}{10-1}}$$

$$= 4,634683$$

$$\sigma_{Z(KT)} = \sqrt{\frac{\sum_{i=1}^n (Z_i - \bar{x}_{Z(KT)})^2}{\text{Jumlah jenis Silhouette KT} - 1}}$$

$$= \sqrt{\frac{(21,33 - 14,734)^2 + (22,69 - 14,734)^2 + (6,61 - 14,734)^2 + (23,14 - 14,734)^2 + (8,97 - 14,734)^2 + (10,93 - 14,734)^2 + (6,71 - 14,734)^2 + (21,03 - 14,734)^2 + (1,13 - 14,734)^2 + (24,8 - 14,734)^2}{10-1}}$$

$$= 8,702342$$

Perhitungan yang sama dilakukan untuk mencari Standar Deviasi dari jenis *Silhouette* KTT, ATT, AT, BTT, BT, DTT, dan DT. Hasil perhitungan Standar Deviasi keseluruhan jenis *Silhouette* ditunjukkan pada Tabel 5.4.

Tabel 5.4 Standar Deviasi masing-masing *Silhouette*

	Berat	X	Y	Z
AT	1,850616	17,61334	9,819413	13,87587
ATT	1,507012	0,240139	0,258683	0,242899
KT	1,27207	15,4507	4,634683	8,702342
KTT	2,011621	0,303864	0,161245	0,438463
BT	3,736356	6,563034	17,91321	11,50133
BTT	0,752762	0,159028	0,31194	0,525167
DT	4,648094	14,55052	10,35896	4,780723
DTT	4,628718	0,240428	0,131339	0,07746

Pada Tabel 5.4 dibawah ini merepresentasikan nilai standar deviasi masing-masing fitur (data sensor) terhadap masing-masing kelas.

Standar deviasi akan diinisialisasikan pada kode program sebagai global variabel, yang berguna untuk perhitungan *Gaussian* dan *posterior* pada program.

2. Menghitung nilai *Gaussian* dari masing-masing fitur

$$\begin{aligned} P(\text{berat} = 38,8 \mid \text{KT}) &= \frac{1}{\sqrt{2\pi\sigma_{\text{berat(KT)}}^2}} e^{-\frac{(38,8 - \bar{x}_{\text{berat(KT)}})^2}{2\sigma_{\text{berat(KT)}}^2}} \\ &= \frac{1}{\sqrt{2 \times 3,14 \times (1,27207)^2}} e^{-\frac{(38,8 - 35,735)^2}{2 \times (1,27207)^2}} \\ &= 0,017213225 \end{aligned}$$

$$\begin{aligned} P(X = -22,54 \mid \text{KT}) &= \frac{1}{\sqrt{2\pi\sigma_{X(\text{KT})}^2}} e^{-\frac{(-22,54 - \bar{x}_{X(\text{KT})})^2}{2\sigma_{X(\text{KT})}^2}} \\ &= \frac{1}{\sqrt{2 \times 3,14 \times (15,4507)^2}} e^{-\frac{(-22,54 + 5,082)^2}{2 \times (15,4507)^2}} \\ &= 0,013640732 \end{aligned}$$

$$\begin{aligned} P(Y = 3,24 \mid \text{KT}) &= \frac{1}{\sqrt{2\pi\sigma_{Y(\text{KT})}^2}} e^{-\frac{(3,24 - \bar{x}_{Y(\text{KT})})^2}{2\sigma_{Y(\text{KT})}^2}} \\ &= \frac{1}{\sqrt{2 \times 3,14 \times (4,634683)^2}} e^{-\frac{(3,24 - 2,007)^2}{2 \times (4,634683)^2}} \\ &= 0,083105792 \end{aligned}$$

$$\begin{aligned} P(Z = 34,6 \mid \text{KT}) &= \frac{1}{\sqrt{2\pi\sigma_{Z(\text{KT})}^2}} e^{-\frac{(34,6 - \bar{x}_{Z(\text{KT})})^2}{2\sigma_{Z(\text{KT})}^2}} \\ &= \frac{1}{\sqrt{2 \times 3,14 \times (8,702342)^2}} e^{-\frac{(34,6 - 14,734)^2}{2 \times (8,702342)^2}} \\ &= 0,003386555 \end{aligned}$$

Perhitungan yang sama dilakukan untuk mencari *Gaussian* dari jenis *Sillhouette* KTT, ATT, AT, BTT, BT, DTT, dan DT. Hasil perhitungan *Gaussian* keseluruhan jenis *Sillhouette* ditunjukkan pada Tabel 5.5.

Tabel 5.5 Gaussian masing-masing *Sillhouette*

	Berat	X	Y	Z
AT	0,205293224	0,016232225	0,014532542	0,003614103
ATT	0,255535915	0,386871975	0,056652012	0,36843487
KT	0,017213225	0,013640732	0,083105792	0,003386555
KTT	0,068079286	0,226073245	0,009081209	0,008237304
BT	0,105577482	0,002393038	0,020748925	0,034430121
BTT	0,521649371	1,256874266	0,534751745	8,11766
DT	0,06300887	0,012630604	0,033944981	0,002546619
DTT	0,039703629	0,167384252	0,012607089	0,042396601

Pada Tabel 5.3 dibawah ini merepresentasikan nilai *gaussian* masing-masing fitur (data sensor) terhadap masing-masing kelas.

3. Menghitung nilai *Posterior* dari masing-masing *Sillhouette*

$$\begin{aligned}
 P(KT \mid \text{berat}, X, Y, Z) &= P_{\text{Kalkun Tertembak}} \times P(\text{berat} = 38,8 \mid KT) \times \\
 &\quad P(X = -22,54 \mid KT) \times P(Y = 3,24 \mid KT) \times P(Z = 34,6 \mid KT) \\
 &= 0,125 \times 0,017213225 \times 0,013640732 \times \\
 &\quad 0,083105792 \times 0,003386555 \\
 &= 8,26037
 \end{aligned}$$

Perhitungan yang sama dilakukan untuk mencari *Posterior* dari jenis *Sillhouette* KTT, ATT, AT, BTT, BT, DTT, dan DT. Perhitungan dilakukan dengan data uji yang berbeda, seperti yang tercantum pada Tabel 5.6 dibawah ini :

Tabel 5.6 Sampel data uji masing-masing *Sillhouette*

	Berat	X	Y	Z
AT	28,70	-20,43	10,78	47,12

	Berat	X	Y	Z
ATT	28,70	13,19	2,04	5,05
KT	38,80	-22,54	3,24	34,60
KTT	38,80	12,89	2,94	3,84
BT	48,10	14,10	24,95	1,73
BTT	48,10	13,19	1,73	2,64
DT	62,30	-30,68	-2,19	-11,99
DTT	62,30	13,04	2,04	2,34

Hasil perhitungan *Posterior* keseluruhan jenis *Sillhouette* ditunjukkan pada Tabel 5.7. Pada Tabel 5.7 dibawah ini merepresentasikan nilai *posterior* masing-masing kelas.

Tabel 5.7 *Posterior* masing-masing *Sillhouette*

<i>Posterior</i>	
AT	2,18779
ATT	3,49317
KT	8,26037
KTT	7,33598
BT	2,25613
BTT	1,30771
DT	8,59954
DTT	1,45429

5.2 Implementasi Sistem

Pada sub-bab ini, akan dijelaskan tentang implementasi sistem pada perangkat keras (*Hardware*) maupun perangkat lunak (*Software*).

5.2.1 Implementasi *Prototype Scoring System*

Pada tahap implementasi *Prototype Scoring System*, alat yang digunakan antara lain *box plastic* yang berfungsi sebagai tempat sensor *Load Cell*, *Switch*, dan *Limit Switch*. Penopang *Sillhouette* berbahan akrilik yang berfungsi selain menjadi penopang *Sillhouette* juga sebagai tempat ditempelnya sensor *ADXL335*. Kontroler ditutupi dengan kardus berwarna hitam untuk menjaga estetika dari sistem ini. Pada tahap Implementasi *Prototype Scoring System* ini harus sesuai dengan Sub Bab 5.1.1 tentang Perancangan *Prototype Scoring System*. Tampilan *prototype* sistem tampak depan terdapat pada Gambar 5.13.



Gambar 5.13 Implementasi *Prototype* Tampak Depan

Pada Gambar 5.13 di atas, ada 4 *Sillhouette* hewan yang berdiri tegak ditopang oleh penopang yang terbuat dari akrilik. Pada penopang *Sillhouette* juga ditempel sensor *ADXL 335 Accelerometer*. Dibawah itu ada 4 *Box Sillhouette* yang berisi sensor *Load Cell*, *Modul HX711*, dan rangkaian *Debounce*. Terakhir dibagian belakang terdapat 1 kotak yang berisi *Controller* atau pusat Kontrol. Selain itu ada pula dokumentasi *Prototype* tampak atas yang tercantum pada Gambar 5.14.

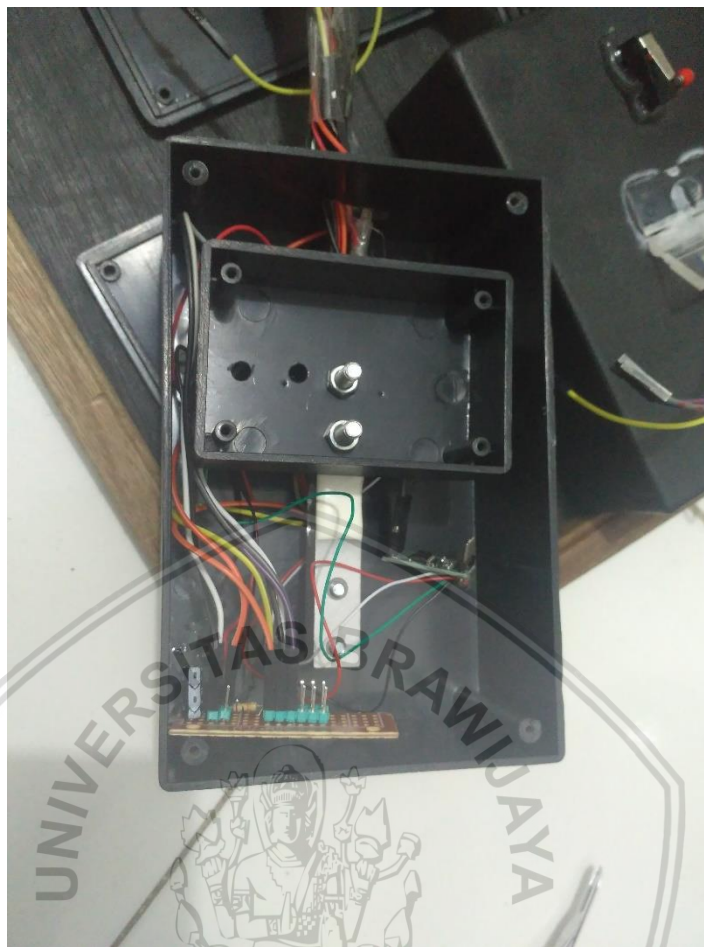


Gambar 5.14 Implementasi *Prototype* Tampak Atas

Pada Gambar 5.14, terdapat 4 *Limit Switch* dan 4 *Switch* yang terpasang pada masing-masing *Box Sillhouette* berwarna hitam.

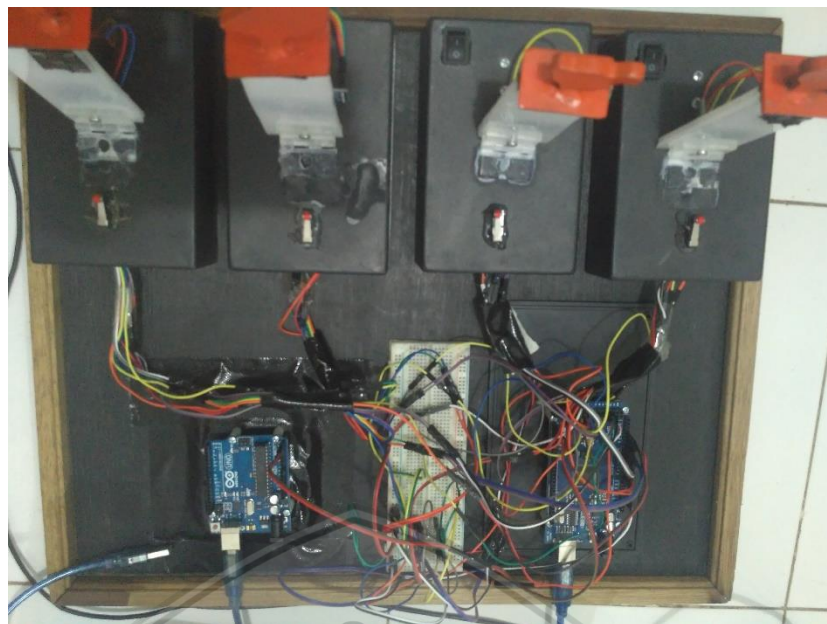
5.2.2 Implementasi Perangkat Keras

Pada tahap implementasi perangkat keras (*Hardware*) dimulailah pemasangan sensor *Accelerometer*, *Load Cell*, *Switch* dan *Limit Switch* pada sistem. *Load Cell* berfungsi untuk mengukur berat pada *Sillhouette*, sensor *Accelerometer* berfungsi untuk mengukur pergerakan pada *Sillhouette* ketika jatuh, *Switch* berfungsi sebagai pengatur kalibrasi manual pada *Load Cell* dan *Limit Switch* berfungsi sebagai trigger pengiriman data *Sillhouette* ke pusat kontroler. Hasil dari implementasi perangkat keras pada *Box Sillhouette* terdapat pada Gambar 5.15.



Gambar 5.15 Implementasi Perangkat Keras pada *Box Silhouette*

Pada Gambar 5.15, terdapat beberapa komponen yang ada pada *Box Silhouette*. Komponen yang berwarna *Silver* adalah sensor *Load Cell*. Pada *Box Silhouette* juga terdapat papan rangkaian *Debounce* yang berwarna coklat. Selain implementasi perangkat keras pada *Box Silhouette*, ada Gambar 5.16 yang merupakan hasil dari implementasi perangkat keras pada Kontroler.



Gambar 5.16 Implementasi Perangkat Keras pada *Controller*

Pada Gambar 5.16, terlihat beberapa komponen yang dipakai seperti *Arduino Uno*, *Arduino Mega 2560*, dan *Project Board*. Pada gambar diatas, kabel *Jumper* lebih banyak terpasang pada *Arduino Mega 2560* karena komponen tersebut digunakan sebagai tempat akuisisi data seluruh sensor. Data sensor kemudian dikirimkan ke *Arduino Uno* menggunakan komunikasi *serial* untuk diklasifikasi. Sedangkan *Project Board* diatas merupakan wadah untuk rangkaian *Debounce*.

5.2.3 Implementasi Perangkat Lunak

Pada tahap implementasi perangkat lunak (*Software*) terdapat 3 sub, yakni Implementasi Kode Program Pembacaan Data Sensor, Implementasi Klasifikasi *Naïve Bayes* pada *Scoring System* dan Implementasi *Scoreboard*. Untuk implementasi perangkat lunak yang pertama kali dilakukan adalah menginisialisasi library yang digunakan pada Arduino IDE. Library yang digunakan pada sistem ini yaitu library *HX711*, *I2Cdev* dan *wire.h*. Untuk programnya tertera pada Tabel 5.8.

Tabel 5.8 Kode program inialisasi library pada sistem

Baris	Kode Program
1	#include "HX711.h"
2	#include "I2Cdev.h"
3	#include <Wire.h>

5.2.3.1 Implementasi Perangkat Lunak Pembacaan Data Sensor

Dalam melakukan implementasi untuk membaca data sensor diperlukan inialisasi variabel dan konfigurasi pin sensor terlebih dahulu. Pada Tabel 5.9, baris pertama menunjukkan inialisasi delay perpindahan silhouette, baris kedua hingga 18 menunjukkan inialisasi variabel dan konfigurasi pin *Load Cell* serta factor kalibrasinya. Pada baris 19 sampai 42 merupakan kode program inialisasi variabel dan konfigurasi pin *Switch* dan *Limit Switch*. Baris 43 sampai 58 merupakan kode program inialisasi variabel dan konfigurasi pin dari sensor *Accelerometer*. Pada baris 59 hingga 66 merupakan kode program inialisasi global variabel.

Tabel 5.9 Kode program inialisasi variabel pembacaan sensor

Baris	Kode Program
1	uint32_t period = 0.1 * 60000L; // 6 s //Load Cell1
2	float calibration_factor = 2230; // this calibration factor is adjusted according to my <i>Load Cell</i> //----Load Cell no.1
3	HX711 scale1(25, 24);
4	float units1;
5	float ounces1;
6	float out1; //----Load Cell no.2
7	HX711 scale2(33, 32);
8	float units2;
9	float ounces2;
10	float out2; //----Load Cell no.3
11	HX711 scale3(41, 40);
12	float units3;

Baris	Kode Program
13	float ounces3;
14	float out3;
	//----Load Cell no.4
15	HX711 scale4(49, 48);
16	float units4;
17	float ounces4;
18	float out4;
	//Limit Switch1
19	const int limit1=23;
20	int LimitState1=0;
21	int nilai_limit1=0;
22	String limit_kon1="";
23	int statePush1=22;
24	int buttonState1;
	//Limit Switch2
25	const int limit2=31;
26	int LimitState2=0;
27	int nilai_limit2=0;
28	String limit_kon2="";
29	int statePush2=30;
30	int buttonState2;
	//Limit Switch3
31	const int limit3=39;
32	int LimitState3=0;
33	int nilai_limit3=0;
34	String limit_kon3="";
35	int statePush3=38;
36	int buttonState3;
	//Limit Switch4
37	const int limit4=47;
38	int LimitState4=0;
39	int nilai_limit4=0;
40	String limit_kon4="";
41	int statePush4=46;
42	int buttonState4;
	//Sensor ACC 1
43	const int xpin1=0,ypin1=1,zpin1=2;
44	float x1=0,y1=0,z1=0;
45	int read_x1=0,read_y1=0,read_z1=0;
46	String nilail="";

Baris	Kode Program
	<pre> //Sensor ACC 2 47 const int xpin2=3,ypin2=4,zpin2=5; 48 float x2=0,y2=0,z2=0; 49 int read_x2=0,read_y2=0,read_z2=0; 50 String nilai2=""; //Sensor ACC 3 51 const int xpin3=8,ypin3=7,zpin3=6; 52 float x3=0,y3=0,z3=0; 53 int read_x3=0,read_y3=0,read_z3=0; 54 String nilai3=""; //Sensor ACC 4 55 const int xpin4=9,ypin4=10,zpin4=11; 56 float x4=0,y4=0,z4=0; 57 int read_x4=0,read_y4=0,read_z4=0; 58 String nilai4=""; //global variable 59 String outx1,outy1,outz1; 60 String outx2,outy2,outz2; 61 String outx3,outy3,outz3; 62 String outx4,outy4,outz4; 63 String acc1,acc2,acc3,acc4; 64 String accelero1,accelero2,accelero3,accelero4; 65 String berat1,berat2,berat3,berat4; 66 String outs1,outs2,outs3,outs4; </pre>

Setelah inisialisasi variabel dan konfigurasi pin, langkah selanjutnya adalah implementasi kode program untuk pembacaan nilai sensor yang dipakai oleh sistem. Tabel 5.10 yang berisi kode program pembacaan nilai state push atau biasa disebut *Switch*. Sistem ini menggunakan 4 state push dengan pembacaan data digital.

Tabel 5.10 Kode program pembacaan nilai state push

Baris	Kode Program
1	<code>buttonState1 = digitalRead(statePush1);</code>
2	<code>buttonState2 = digitalRead(statePush2);</code>
3	<code>buttonState3 = digitalRead(statePush3);</code>
4	<code>buttonState4 = digitalRead(statePush4);</code>

Selain state push atau *Switch*, ada sensor *Limit Switch* yang berfungsi sebagai trigger pengiriman data. Tabel 5.11 yang berisi kode program pembacaan nilai *Limit Switch*. Sama seperti state push atau *Switch*, *Limit Switch* yang digunakan berjumlah 4 buah dengan pembacaan data digital.

Tabel 5.11 Kode program pembacaan nilai *Limit Switch*

Baris	Kode Program
1	<code>LimitState1 = digitalRead(limit1);</code>
2	<code>LimitState2 = digitalRead(limit2);</code>
3	<code>LimitState3 = digitalRead(limit3);</code>
4	<code>LimitState4 = digitalRead(limit4);</code>

Sensor *Accelerometer* atau biasa disebut seri *ADXL335* memiliki 3 *output* analog yakni X, Y dan Z. Pada Tabel 5.12, baris 1, baris 14, baris 26 dan baris 39 merupakan inisialisasi masing-masing *method*. Pada baris 2 hingga 4 merupakan kode program pembacaan nilai sensor *Accelerometer* dengan data analog. *Output* sensor ini berupa angka dengan tipe data float yang memiliki angka dibelakang koma yang banyak. Diperlukan kalibrasi agar data yang diperoleh lebih akurat. Pada baris 5 sampai baris 7 merupakan kode program implementasi kalibrasi data *Accelerometer*. *Output* dari sensor *Accelerometer* yang berupa data float diubah ke tipe data string untuk mempermudah penyatuan data sensor yang lain. Baris 9 sampai baris 11 merupakan kode program implementasi perubahan tipe data sensor *Accelerometer*. Kemudian data disatukan ke variabel *outs1*, *outs2*, *outs3* dan *outs4*. Kode program implementasinya terdapat pada baris 12, baris 24, baris 37 dan baris 51.

Tabel 5.12 Kode program pembacaan nilai sensor *Accelerometer*

Baris	Kode Program
1	String Read_Acc1() {
2	read_x1=analogRead(xpin1);
3	read_y1=analogRead(ypin1);
4	read_z1=analogRead(zpin1);
5	x1=((float) read_x1-331.5)/65*9.8;
6	y1=((float) read_y1-331.5)/65*9.8;
7	z1=((float) read_z1-331.5)/65*9.8;
8	String outx1,outy1,outz1,coma1=",";
9	outx1=String(x1);
10	outy1=String(y1);
11	outz1=String(z1);
12	String outs1=outx1+coma1+outy1+coma1+outz1;
13	return outs1;}
14	String Read_Acc2() {
15	read_x2=analogRead(xpin2);
16	read_y2=analogRead(ypin2);
17	read_z2=analogRead(zpin2);
18	x2=((float) read_x2-331.5)/65*9.8;
19	y2=((float) read_y2-331.5)/65*9.8;
20	z2=((float) read_z2-331.5)/65*9.8;
21	String outx2,outy2,outz2,coma2=",";
22	outx2=String(x2);
23	outy2=String(y2);
23	outz2=String(z2);
24	String outs2=outx2+coma2+outy2+coma2+outz2;
25	return outs2;}
26	String Read_Acc3() {
27	read_x3=analogRead(xpin3);
28	read_y3=analogRead(ypin3);
29	read_z3=analogRead(zpin3);
30	x3=((float) read_x3-331.5)/65*9.8;
31	y3=((float) read_y3-331.5)/65*9.8;
32	z3=((float) read_z3-331.5)/65*9.8;
33	String outx3,outy3,outz3,coma3=",";
34	outx3=String(x3);

Baris	Kode Program
35	<code>outy3=String(y3);</code>
36	<code>outz3=String(z3);</code>
37	<code>String outs3=outx3+coma3+outy3+coma3+outz3;</code>
38	<code>return outs3;}</code>
39	<code>String Read_Acc4() {</code>
40	<code> read_x4=analogRead(xpin4);</code>
41	<code> read_y4=analogRead(ypin4);</code>
42	<code> read_z4=analogRead(zpin4);</code>
43	<code> x4=((float)read_x4-331.5)/65*9.8;</code>
44	<code> y4=((float)read_y4-331.5)/65*9.8;</code>
45	<code> z4=((float)read_z4-331.5)/65*9.8;</code>
46	<code> String outx4,outy4,outz4,coma4=",";</code>
47	<code> outx4=String(x4);</code>
48	<code> outy4=String(y4);</code>
49	<code> outz4=String(z4);</code>
50	<code> String outs4=outx4+coma4+outy4+coma4+outz4;</code>
51	<code> return outs4;}</code>

Sensor terakhir yang dipakai oleh sistem adalah sensor *Load Cell*. Peran sensor *Load Cell* sangat bergantung pada state push atau *Switch*, karena peletakan sensor ini yang perlu dikalibrasi setiap sistem akan dijalankan dan *Switch* membantu dalam proses kalibrasi itu. Cara kerjanya yakni dimulai dengan memberi nilai 0 pada state push, lalu state push pada *box* pertama diberi nilai 1, setelah itu kalibrasi posisi *Load Cell* secara manual. Begitu seterusnya untuk *box-box* selanjutnya. Kode program pembacaan nilai sensor ini tercantum pada Tabel 5.13.

Tabel 5.13 Kode program pembacaan nilai sensor *Load Cell*

Baris	Kode Program
1	<code>scale1.set_scale();</code>
2	<code>scale1.tare(); //Reset the scale to 0</code>
3	<code>buttonState1 = digitalRead(statePush1);</code>
4	<code>scale2.set_scale();</code>
5	<code>scale2.tare(); //Reset the scale to 0</code>

Baris	Kode Program
6	<code>buttonState2 = digitalRead(statePush2);</code>
7	<code>scale3.set_scale();</code>
8	<code>scale3.tare(); //Reset the scale to 0</code>
9	<code>buttonState3 = digitalRead(statePush3);</code>
10	<code>scale4.set_scale();</code>
11	<code>scale4.tare(); //Reset the scale to 0</code>
12	<code>buttonState4 = digitalRead(statePush4);</code>
13	<code>Serial.println("Inisialisasi");</code>
14	<code>while(buttonState1!=1 and buttonState2!=1 and buttonState3!=1 and buttonState4!=1){</code>
15	<code>buttonState1 = digitalRead(statePush1);</code>
16	<code>buttonState2 = digitalRead(statePush2);</code>
17	<code>buttonState3 = digitalRead(statePush3);</code>
18	<code>buttonState4 = digitalRead(statePush4);</code>
19	<code>while(buttonState1!=1){</code>
20	<code>delay(5000);</code>
21	<code>for(int i=0;i<50;i++){</code>
22	<code>Serial.print("Masuk Kesini 1 : ");</code>
23	<code>Serial.println(out1);</code>
24	<code>scale1.set_scale(calibration_factor); //Adjust to this calibration factor</code>
25	<code>units1= scale1.get_units(), 10;</code>
26	<code>if (units1 < 5){</code>
27	<code>units1 = 0.00;}</code>
28	<code>ounces1 = units1 * 0.035274;</code>
29	<code>out1=units1*6.5;</code>
30	<code>}Serial.print("Sillouette 1 :</code>
31	<code>");Serial.println(out1);buttonState1 = digitalRead(statePush1); }</code>
32	<code>while(buttonState2!=1){</code>
33	<code>delay(5000);</code>
34	<code>for(int i=0;i<50;i++){</code>
35	<code>Serial.print("Masuk Kesini 2 : ");</code>
36	<code>Serial.println(out2);</code>
37	<code>scale2.set_scale(calibration_factor); //Adjust to this calibration factor</code>
38	<code>units2= scale2.get_units(), 10;</code>
39	<code>if (units2 < 5){</code>

Baris	Kode Program
40	units2 = 0.00;}
41	ounces2 = units2 * 0.035274;
42	out2=units2*6.5;
43	}Serial.print("Sillhouette 2 : ");Serial.println(out2);
44	buttonState2 = digitalRead(statePush2);
	}
45	while(buttonState3!=1){
46	delay(5000);
47	for(int i=0;i<50;i++){
48	Serial.print("Masuk Kesini 3 : ");
49	Serial.println(out3);
50	scale3.set_scale(calibration_factor); //Adjust to this calibration factor
51	units3= scale3.get_units(), 10;
52	if (units3 < 5){
53	units3 = 0.00; }
54	ounces3 = units3 * 0.035274;
55	out3=units3*6.5;
56	}Serial.print("Sillhouette 3 : ");Serial.println(out3);
57	buttonState3 = digitalRead(statePush3);
	}
58	while(buttonState4!=1){
59	delay(5000);
60	for(int i=0;i<50;i++){
61	Serial.print("Masuk Kesini 4 : ");
62	Serial.println(out4);
63	scale4.set_scale(calibration_factor); //Adjust to this calibration factor
64	units4= scale4.get_units(), 10;
65	if (units4 < 5){
66	units4 = 0.00; }
67	ounces4 = units4 * 0.035274;
68	out4=units4*6.5;
69	}Serial.print("Sillhouette 4 : ");Serial.println(out4);
70	buttonState4 = digitalRead(statePush4); } }
71	Serial.println("KELUAR BOSQU!!!");
72	delay(1000);}

5.2.3.2 Implementasi Klasifikasi *Naïve Bayes* pada *Scoring System*

Pada tahap ini, dilakukan proses komputasi data sensor menggunakan Klasifikasi *Naïve Bayes*. Seperti langkah-langkah sebelumnya, pada tahap implementasi klasifikasi *Naïve Bayes* dimulai dengan inisialisasi variabel *Naïve Bayes* pada kode program. Tabel 5.14 merupakan kode program inisialisasi variabel dan tipe data yang digunakan pada metode *Naïve Bayes*.

Tabel 5.14 Kode program inisialisasi variabel dan tipe data metode *Naïve Bayes*

Baris	Kode Program
1	<code>float gaussian[3][4];</code>
2	<code>float hasil[3];</code>
3	<code>float tertinggi = -1.000; //set default</code>
4	<code>int index = 0;</code>
5	<code>int gauske = 0;</code>

Proses komputasi metode *Naïve Bayes* dimulai dari perhitungan peluang *prior* masing-masing kelas. Pada sistem ini terdapat 8 kelas, yakni Ayam Tertembak (AT), Ayam Tidak Tertembak (ATT), Kalkun Tertembak (KT), Kalkun Tidak Tertembak (KTT), Babi Tertembak (BT), Babi Tidak Tertembak (BTT), Domba Tertembak (DT), dan Domba Tidak Tertembak (DTT). Setelah itu dilakukan perhitungan *mean* dan standar deviasi, hingga dapat hasil perhitungannya sesuai dengan sub bab 5.1.3.2. Hasil perhitungan peluang *prior*, *mean*, dan standar deviasi diimplementasikan kedalam kode program seperti pada Tabel 5.15. Pada baris 1 hingga baris 8 adalah inisialisasi variabel, tipe data dan nilai dari peluang *prior* tiap kelas. Baris 9 hingga baris 16 adalah inisialisasi variabel, tipe data, dan nilai dari *mean* dan standar deviasi.

Tabel 5.15 Kode program inisialisasi variabel nilai *Prior*, *Mean*, & Standar Deviasi

Baris	Kode Program
1	<code>float pATT = 0.125; //pATT = 1 / 8;</code>
2	<code>float pAT = 0.125; //pAT = 1 / 8;</code>
3	<code>float pKTT = 0.125; //pKTT = 1 / 8;</code>
4	<code>float pKT = 0.125; //pKT = 1 / 8;</code>

Baris	Kode Program
5	float pBTT = 0.125; //pBTT = 1 / 8;
6	float pBT = 0.125; //pBT = 1 / 8;
7	float pDTT = 0.125; //pDTT = 1 / 8;
8	float pDT = 0.125; //pDT = 1 / 8;
9	float ATT[2][4] = {{30.141, 11.906, 0.455, 4.194}, {2.975288782, 0.295115796, 0.257390754, 0.294437316}}; // {{mean1, mean2, mean3, mean4}, {sd1, sd2, sd3, sd4}};
10	float AT[2][4] = {{28.12, 3.3, 0.545, 14.1}, {1.850615513, 0.202484567, 0.239269722, 0.122474487}}; // {{mean1, mean2, mean3, mean4}, {sd1, sd2, sd3, sd4}};
11	float KTT[2][4] = {{35.722, 11.654, 0.8, 2.518}, {2.094064628, 0.225743217, 0.281069386, 0.649988034}}; // {{mean1, mean2, mean3, mean4}, {sd1, sd2, sd3, sd4}};
12	float KT[2][4] = {{35.735, 3.18, 0.74, 13.13}, {1.272069617, 0.236643191, 0.225831796, 0.104880885}}; // {{mean1, mean2, mean3, mean4}, {sd1, sd2, sd3, sd4}};
13	float BTT[2][4] = {{46.97, 11.53, 1.135, 1.49}, {2.648232115, 0.122474487, 0.286210878, 0.99589156}}; // {{mean1, mean2, mean3, mean4}, {sd1, sd2, sd3, sd4}};
14	float BT[2][4] = {{47.533, 3.63, 1.205, 11.215}, {3.736356068, 0.176068169, 0.190394328, 0.16507574}}; // {{mean1, mean2, mean3, mean4}, {sd1, sd2, sd3, sd4}};
15	float DTT[2][4] = {{68.064, 11.53, 0.755, 1.597}, {4.628717605, 0.158113883, 0.302076149, 0.417374066}}; // {{mean1, mean2, mean3, mean4}, {sd1, sd2, sd3, sd4}};
16	float DT[2][4] = {{68.205, 2.67, 1.13, 11.729}, {5.321385054, 0.32249031, 0.353553391, 0.217278623}}; // {{mean1, mean2, mean3, mean4}, {sd1, sd2, sd3, sd4}};

Selanjutnya implementasi kode program untuk komputasi fungsi *gaussian* berdasarkan yang telah dirancang pada Gambar 5.8. Kode program fungsi *gaussian* tertera pada Tabel 5.16. Pada baris ke-1 merupakan inialisasi fungsi *gaussian* dimana parameter perhitungannya ditentukan oleh suatu variabel array berukuran 1x4 yang merepresentasikan nilai pembacaan sensor dan parameter lainnya adalah variabel array berukuran 2x4 yang merepresentasikan nilai olahan data latih. Kemudian pada baris ke-2 menunjukkan inialisasi variabel yang akan

digunakan untuk menentukan rumus perhitungan. Lalu pada baris ke-3 hingga baris ke-8 menunjukkan proses perhitungan *gaussian* secara berulang-ulang hingga keseluruhan fitur dari pembacaan sensor didapatkan peluang *gaussiannya* masing-masing. Pada baris ke-9 melakukan iterasi sebanyak satu kali untuk variabel *gauske*.

Tabel 5.16 Kode program fungsi *gaussian*

Baris	Kode Program
1	<code>void gaussian(float aq[4], float base[2][4]) {</code>
2	<code>double d, e, f, g;</code>
3	<code>for (int i = 0; i < 4; i++) {</code>
4	<code> d = 2 * 3.14 * (pow(base[1][i], 2));</code>
5	<code> e = -((pow((aq[i] - base[0][i]), 2)) / (2 * pow(base[1][i],</code>
	<code>2))));</code>
6	<code> f = pow(2.718282, e);</code>
7	<code> g = 1 / sqrt(d);</code>
8	<code> gaussian[gauske][i] = f * g; }</code>
9	<code> gauske++; }</code>

Tabel 5.17 menunjukkan implementasi kode pemrograman untuk mendapatkan nilai hasil peluang *Sillhouette* terjatuh dari data yang diujikan. Pada baris ke-1 menunjukkan inisialisasi fungsi *ProbPosterior* dengan parameter perhitungannya adalah peluang dari masing-masing kelas dan nilai *gaussian* dari masing-masing fitur. Selanjutnya pada baris ke-2 hingga baris ke-6 menunjukkan perulangan untuk melakukan perkalian antar nilai *gaussian* keseluruhan fitur. Selanjutnya pada baris ke-7, hasil perkalian antar nilai *gaussian* keseluruhan fitur di kalikan lagi dengan peluang *prior* sehingga didapatkan nilai peluang *posterior*.

Tabel 5.17 Kode program fungsi *ProbPosterior*

Baris	Kode Program
1	<code>void ProbPosterior(float prior, int i) {</code>
2	<code> for (int j = 0; j < 4; j++) {</code>
3	<code> if (j == 0) {</code>
4	<code> hasil[i] = (gaussian[i][j] * 1000); // dikali 1000 agar</code>
	<code> angka di belakang koma tidak hilang</code>
	<code> }</code>

Baris	Kode Program
5	else {
6	hasil[i] = hasil[i] * (gaussian[i][j] * 1000);
	}
	}
7	hasil[i] = hasil[i] * prior; }

Langkah terakhir pada implementasi Klasifikasi *Naïve Bayes* adalah menarik kesimpulan dari hasil klasifikasi *Sillhouette* yang dilakukan dengan membandingkan antar nilai peluang *posterior* mana yang mempunyai nilai tertinggi seperti ditunjukkan pada Tabel 5.18. Terlihat pada baris ke-2 hingga baris ke-8 setiap nilai peluang *posterior* dari jenis dehidrasi dibandingkan, kemudian nilai yang tertinggi menandakan data uji tersebut termasuk kedalam jenis *Sillhouette* dari peluang *posterior* yang di maksud. Selanjutnya pada baris ke-9 hingga baris ke-36 menampilkan jenis *Sillhouette* yang sesuai dengan hasil perbandingan.

Tabel 5.18 Kode program penarikan kesimpulan

Baris	Kode Program
1	void kesimpulan() {
2	for (int i = 0; i < 8; i++) {
3	if (i == 0){
4	tertinggi = hasil[i];
5	index = i + 1;
	}
6	else if (tertinggi < hasil[i]){
7	tertinggi = hasil[i];
8	index = i + 1; } }
9	Serial.print("Terdeteksi :");
10	if (index == 1) {
11	Serial.print("Ayam Normal : ");
12	Serial.print(skor); }
13	else if (index == 2) {
14	Serial.print("Ayam Tertembak : ");
15	Serial.print(skor); }
16	else if (index == 3) {

Baris	Kode Program
17	Serial.print("Kalkun Normal : ");
18	Serial.print(skor); }
19	else if (index == 4) {
20	skor += 10;
22	Serial.print("Kalkun Tertembak : ");
23	Serial.print(skor); }
24	else if (index == 5) {
25	Serial.print("Babi Normal : ");
26	Serial.print(skor); }
27	else if (index == 6) {
28	Serial.print("Babi Tertembak : ");
29	Serial.print(skor); }
30	else if (index == 7) {
31	Serial.print("Domba Normal : ");
32	Serial.print(skor); }
33	else if (index == 8) {
34	Serial.print("Domba Tertembak : ");
35	Serial.print(skor); }
36	delay(1000); }



BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas proses pengujian serta menganalisis hasil dari pengujian yang dilakukan berdasarkan sistem yang telah dibuat. Adapun tujuan dilakukannya pengujian ada untuk mengetahui apakah semua kebutuhan yang diharapkan telah terpenuhi oleh sistem. Terdapat 4 pengujian yang akan dilakukan, yaitu pengujian sensor *ADXL335 Accelerometer*, *Load Cell*, *Limit Switch*, dan proses klasifikasi *Naïve Bayes*.

6.1 Pengujian Sensor *Load Cell*

6.1.1 Tujuan Pengujian

Pengujian sensor *Load Cell* memiliki tujuan, yakni untuk mengetahui berat dari masing-masing jenis *Silhouette*.

6.1.2 Prosedur Pengujian

Prosedur yang harus dilakukan untuk melakukan pengujian pada *Load Cell* antara lain :

- 1) Hubungkan pin sensor *Load Cell* dengan *Arduino Mega 2560* menggunakan kabel *jumper* sesuai pada Tabel 5.1. Contoh *Wiring* pada sensor *Load Cell* terdapat pada Gambar 6.1.



Gambar 6.1 Wiring sensor Load Cell

- 2) Buka Arduino IDE dan memprogramnya untuk membaca nilai berat pada *Load Cell*. Untuk kode program tertera pada Tabel 5.13.
- 3) Compile dan upload kode program yang telah dibuat.
- 4) Memberikan *input* pada *Load Cell* berupa jenis *Sillhouette* yang berbeda-beda yaitu *Sillhouette* jenis ayam, kalkun, babi, dan domba.
- 5) Amati hasil *output* pada *Load Cell* yang ditampilkan pada serial monitor Arduino IDE. Kemudian mengambil 40 sampel dari *Load Cell* dan timbangan. Contoh *output* dari *Load Cell* yang tertera pada *Serial Monitor* terdapat pada Gambar 6.2.

```
Load Cell 2 : 22.13 grams calibration_factor: 1130.00
Load Cell 2 : 22.68 grams calibration_factor: 1130.00
Load Cell 2 : 24.53 grams calibration_factor: 1130.00
Load Cell 2 : 26.16 grams calibration_factor: 1130.00
Load Cell 2 : 27.33 grams calibration_factor: 1130.00
Load Cell 2 : 26.30 grams calibration_factor: 1130.00
Load Cell 2 : 26.07 grams calibration_factor: 1130.00
Load Cell 2 : 25.96 grams calibration_factor: 1130.00
Load Cell 2 : 26.76 grams calibration_factor: 1130.00
Load Cell 2 : 26.56 grams calibration_factor: 1130.00
Load Cell 2 : 26.10 grams calibration_factor: 1130.00
Load Cell 2 : 25.15 grams calibration_factor: 1130.00
Load Cell 2 : 24.57 grams calibration_factor: 1130.00
Load Cell 2 : 25.05 grams calibration_factor: 1130.00
Load Cell 2 : 27.54 grams calibration_factor: 1130.00
Sillhouette 2 : 27.54 gramsLoad Cell 2 : 0.00 grams calibration_factor: 1130.00
```

Gambar 6.2 Output sensor Load Cell

- 6) Menentukan Sistem Error (%) dengan rumus seperti **Persamaan (6.1)** dibawah ini :

$$\text{Sistem Error (\%)} = \frac{[PL-PT]}{PL} \times 100 \quad (6.1)$$

Keterangan :

PL : Pembacaan Nilai Pada *Load Cell*

PT : Pembacaan Nilai Pada Timbangan Digital

- 7) Menentukan rata-rata dari Sistem Error(%) dengan rumus seperti **Persamaan (6.2)** dibawah ini :

$$\text{Rata - rata} = \frac{\text{Jumlah Hasil}}{\text{Banyaknya Percobaan}} \quad (6.2)$$

8) Kesimpulan.

6.1.3 Hasil dan Analisis Pengujian

Hasil pengujian sensor *Load Cell* ditunjukkan pada Tabel 6.1.

Tabel 6.1 Hasil Pengujian sensor *Load Cell*

Percobaan ke-	Jenis <i>Silhouette</i>	Akuisisi Data <i>Load Cell</i> (gram)	Akuisisi Data Timbangan Digital (gram)	Persentase Error (%)
1	Babi	48,25	48	0,52
2	Babi	48,3	48	0,62
3	Babi	47,85	48	0,31
4	Babi	46,94	48	2,25
5	Babi	47,12	48	1,86
6	Babi	48,12	48	0,25
7	Babi	49,01	48	2,06
8	Babi	48,98	48	2
9	Babi	48,89	48	1,82
10	Babi	48,89	48	1,82
11	Ayam	29,56	29	1,89
12	Ayam	31,74	29	8,63
13	Ayam	29,93	29	3,11
14	Ayam	27,46	29	5,61
15	Ayam	28,07	29	3,31
16	Ayam	28,04	29	3,42
17	Ayam	27,95	29	3,76
18	Ayam	29,99	29	3,3
19	Ayam	30,81	29	5,87
20	Ayam	27,47	29	5,57
21	Kalkun	37,74	36	4,61
22	Kalkun	36,52	36	1,42

Percobaan ke-	Jenis <i>Silhouette</i>	Akuisisi Data <i>Load Cell</i> (gram)	Akuisisi Data Timbangan Digital (gram)	Persentase Error (%)
23	Kalkun	37,22	36	3,28
24	Kalkun	36,15	36	0,41
25	Kalkun	30,54	36	17,88
26	Kalkun	36,21	36	0,58
27	Kalkun	36,66	36	1,8
28	Kalkun	36,83	36	2,25
29	Kalkun	35,36	36	1,81
30	Kalkun	35,35	36	1,84
31	Domba	59,95	68	13,43
32	Domba	68,49	68	0,72
33	Domba	62,65	68	8,54
34	Domba	53,2	68	27,82
35	Domba	63,21	68	7,58
36	Domba	68,12	68	0,18
37	Domba	64	68	6,25
38	Domba	70,09	68	2,98
39	Domba	74,2	68	8,36
40	Domba	67,22	68	1,16
Rata-rata				4,27

Pada Tabel 6.2 ditunjukkan bahwa rata-rata nilai error pada pengujian sensor *Load Cell* adalah sebesar 4,27%. Hasil tersebut didapatkan setelah melalui berbagai macam prosedur sehingga berhasil didapatkan data dari *Load Cell* dan timbangan digital sebagai parameter pembandingan.

6.2 Pengujian Sensor *ADXL 335 Accelerometer*

6.2.1 Tujuan Pengujian

Tujuan dilakukannya pengujian pada sensor *ADXL335 Accelerometer* adalah untuk mengetahui percepatan dan getaran dari *Sillhouette* ketika *Sillhouette* tertembak. Pada pengujian sensor *ADXL335 Accelerometer* dilakukan dengan 2 perlakuan, yaitu ketika *Sillhouette* dalam keadaan normal (tidak tertembak) dan ketika *Sillhouette* dalam keadaan ditembak.

6.2.2 Prosedur Pengujian

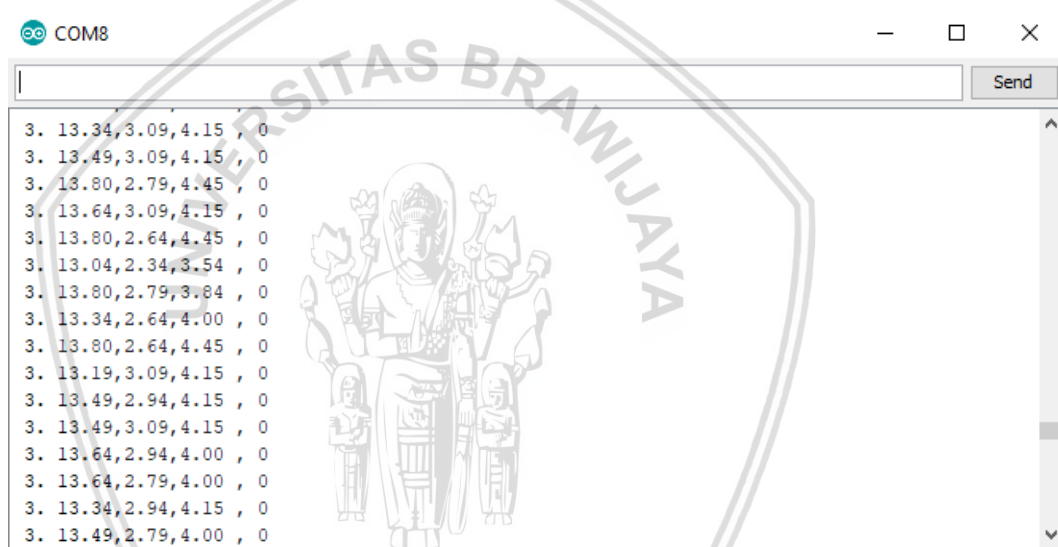
Berikut merupakan prosedur untuk melakukan pengujian pada sensor *ADXL335 Accelerometer* :

- 1) Hubungkan pin sensor *ADXL335 Accelerometer* dengan *Arduino Mega 2560* menggunakan kabel jumper sesuai pada Tabel 5.1. Berikut *Wiring* pada sensor *ADXL 335 Accelerometer* terdapat pada Gambar 6.3.



Gambar 6.3 Wiring sensor ADXL 335 Accelerometer

- 2) Buka Arduino IDE dan tulis kode program untuk membaca sensor *ADXL335 Accelerometer* sesuai pada tabel 5.12.
- 3) Compile dan upload kode program yang telah dibuat.
- 4) Memberikan *input* pada sensor *ADXL335 Accelerometer* berupa 2 perlakuan yaitu, normal (*Silhouette* dalam keadaan berdiri diam) dan ditembak.
- 5) Amati hasil *output* pada sensor *ADXL335 Accelerometer* yang ditampilkan pada serial monitor Arduino IDE, dan mengambil 40 sampel dari hasil pembacaan tersebut. Contoh *output* dari sensor *ADXL 335 Accelerometer* terdapat pada Gambar 6.4.



Gambar 6.4 Output sensor ADXL 335 Accelerometer

- 6) Kesimpulan.

6.2.3 Hasil dan Analisis Pengujian

Hasil pengujian sensor *ADXL335 Accelerometer* ditunjukkan pada Tabel 6.2.

Tabel 6.2 Hasil Pengujian sensor *ADXL335 Accelerometer*

Percobaan ke-	Jenis <i>Silhouette</i>	Kondisi	Nilai sensor <i>ADXL335 Accelerometer</i>		
			X	Y	Z
1	Babi	Normal	13,59	2,19	2,49
2	Babi	Normal	13,53	1,88	2,79
3	Babi	Normal	13,44	2,04	2,49
4	Babi	Normal	13,14	2,19	2,34
5	Babi	Normal	13,29	2,04	2,64
6	Babi	Tertembak	-22,84	19,98	13,04
7	Babi	Tertembak	-22,54	19,08	14,44
8	Babi	Tertembak	-23,43	19,27	13,07
9	Babi	Tertembak	-23,14	16,36	15,45
10	Babi	Tertembak	-22,84	19,98	13,04
11	Ayam	Normal	13,74	1,13	4,45
12	Ayam	Normal	13,59	1,28	4,45
13	Ayam	Normal	13,74	1,58	5,05
14	Ayam	Normal	13,29	1,43	4,75
15	Ayam	Normal	13,89	1,28	4,75
16	Ayam	Tertembak	1,58	5,05	16,66
17	Ayam	Tertembak	1,13	5,05	19,98
18	Ayam	Tertembak	2,34	2,94	9,42
19	Ayam	Tertembak	-36,41	-21,33	37,61
20	Ayam	Tertembak	15,6	2,19	-6,56
21	Kalkun	Normal	13,29	2,34	4,3
22	Kalkun	Normal	13,74	2,34	4,6
23	Kalkun	Normal	13,59	2,64	5,35
24	Kalkun	Normal	13,74	2,04	4,9
25	Kalkun	Normal	13,04	2,49	5,65

Percobaan ke-	Jenis <i>Silhouette</i>	Kondisi	Nilai sensor <i>ADXL335</i> <i>Accelerometer</i>		
			X	Y	Z
26	Kalkun	Tertembak	5,2	8,67	21,33
27	Kalkun	Tertembak	4	8,37	22,69
28	Kalkun	Tertembak	22,69	7,61	6,61
29	Kalkun	Tertembak	3,84	-3,39	23,14
30	Kalkun	Tertembak	-21,48	2,49	8,97
31	Domba	Normal	13,29	2,34	2,64
32	Domba	Normal	13,89	2,64	2,49
33	Domba	Normal	13,44	2,49	2,64
34	Domba	Normal	13,44	2,49	2,64
35	Domba	Normal	13,84	2,34	2,64
36	Domba	Tertembak	0,08	2,49	11,68
37	Domba	Tertembak	4,45	3,24	22,24
38	Domba	Tertembak	-23,29	-0,68	16,81
39	Domba	Tertembak	-22,84	0,05	12,29
40	Domba	Tertembak	-21,18	6,26	16,66

Pada Tabel 6.2 yang merupakan hasil dari pembacaan sensor *ADXL335 Accelerometer* setelah melalui berbagai macam prosedur sehingga berhasil didapatkan data dari Sensor *ADXL335 Accelerometer*. Hal ini menunjukkan bahwa percepatan dan getaran pada setiap jenis *Silhouette* berbeda-beda tergantung dari kondisi perlakuan pada *Silhouette* yaitu normal dan tertembak. Ketika kondisi *Silhouette* normal nilai pada masing-masing sumbu berbeda-beda namun cenderung stabil, karena pada saat kondisi normal *Silhouette* tidak bergerak. Berbeda ketika *Silhouette* tertembak, masing-masing sumbu akan menghasilkan nilai yang berbeda-beda, namun *output* nya cenderung tidak stabil. Hal ini dikarenakan kecepatan peluru ketika berbenturan dengan *Silhouette* tidak stabil

yang mengakibatkan *Sillhouette* terjatuh dengan kecepatan yang berbeda sehingga tak jarang hasil akusisi data sensor ini bernilai minus.

6.3 Pengujian Sensor *Limit Switch*

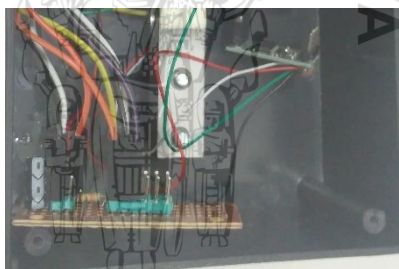
6.3.1 Tujuan Pengujian

Pengujian sensor *Limit Switch* bertujuan untuk mengetahui apakah *Sillhouette* jatuh atau tidak.

6.3.2 Prosedur Pengujian

Prosedur yang harus dilakukan untuk melakukan pengujian pada *Limit Switch* antara lain :

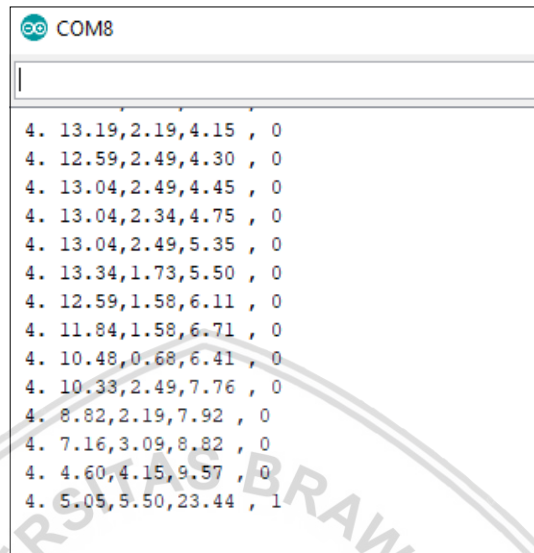
- 1) Merancang *Limit Switch* dengan rangkaian *Debounce* dan pinnya dihubungkan ke *Arduino Mega 2560* menggunakan kabel jumper sesuai pada Tabel 5.1. *Wiring* pada rangkaian *Debounce* terdapat pada Gambar 6.5.



Gambar 6.5 *Wiring* rangkaian *Debounce*

- 2) Buka Arduino IDE dan tulis kode programnya untuk membaca nilai pada *Limit Switch* seperti pada Tabel 5.11.
- 3) *Compile* dan upload kode program yang telah dibuat.
- 4) Memberikan input pada *Limit Switch* berupa *Sillhouette* terjatuh atau tidak.
- 5) Amati hasil *output* pada *Limit Switch* yang ditampilkan pada serial monitor Arduino IDE. Jika nilai *Limit Switch* sama dengan 0, bisa dikatakan *Sillhouette* dalam keadaan normal atau tidak tertembak. Namun jika nilai *Limit Switch* sama dengan 1, bisa dikatakan *Sillhouette* dalam keadaan

terjatuh atau tertembak. Contoh *output* dari nilai *Limit Switch* terdapat pada Gambar 6.6. Angka yang paling kanan merepresentasikan nilai *Limit Switch*.



Gambar 6.6 Output Limit Switch

6) Kesimpulan.

6.3.2 Hasil dan Analisis Pengujian

Hasil pengujian sensor *Limit Switch* ditunjukkan pada Tabel 6.3.

Tabel 6.3 Hasil Pengujian sensor *Limit Switch*

Percobaan ke-	Kondisi	Akuisisi Data Sensor <i>Limit Switch</i>
1	<i>Sillhouette</i> Normal	0
2	<i>Sillhouette</i> Normal	0
3	<i>Sillhouette</i> Normal	0
4	<i>Sillhouette</i> Normal	0
5	<i>Sillhouette</i> Normal	0
6	<i>Sillhouette</i> Terjatuh	1
7	<i>Sillhouette</i> Terjatuh	1
8	<i>Sillhouette</i> Terjatuh	1
9	<i>Sillhouette</i> Terjatuh	1
10	<i>Sillhouette</i> Terjatuh	1

6.4 Pengujian Akurasi Hasil Klasifikasi *Naïve Bayes*

6.4.1 Tujuan Pengujian

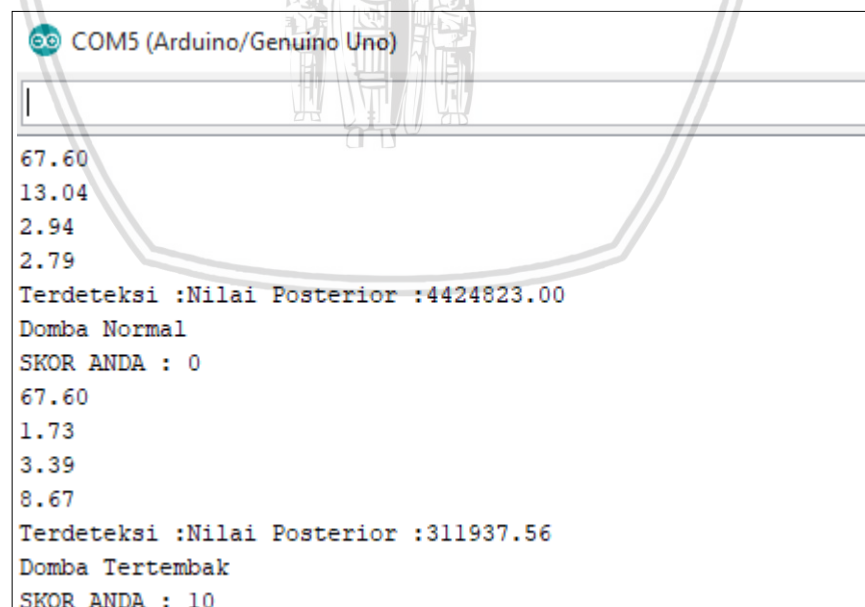
Scoring System Pada Lomba Menembak Dengan Target *Sillhouette* Hewan Menggunakan Metode *Naïve Bayes* ini mempunyai tujuan utama untuk dapat

mengklasifikasikan jenis *Sillhouette* yang tertembak dan bagaimana proses pemberian skornya. Oleh karena itu perlu diketahui tingkat keakuratan sistem dalam melakukan klasifikasi *Naïve Bayes*.

6.4.2 Prosedur Pengujian

Prosedur yang harus dilakukan untuk melakukan pengujian pada Klasifikasi *Naïve Bayes* antara lain :

- 1) Merancang data latih yang diperlukan.
- 2) Buka Arduino IDE dan tulis kode programnya, masukan nilai *mean* dan *standar deviasi* ke dalam kode program seperti pada Tabel 5.14 hingga Tabel 5.17.
- 3) *Compile* dan *upload* kode program yang telah dibuat.
- 4) Buka *Serial Monitor* pada Arduino IDE.
- 5) Tembak *Sillhouette* yang dibidik.
- 6) Amati hasil *output* yang ditampilkan pada *Serial Monitor* Arduino IDE. Contoh *output* dari pengujian akurasi Klasifikasi *Naïve Bayes* terdapat pada Gambar 6.7.



```
COM5 (Arduino/Genuino Uno)
67.60
13.04
2.94
2.79
Terdeteksi :Nilai Posterior :4424823.00
Domba Normal
SKOR ANDA : 0
67.60
1.73
3.39
8.67
Terdeteksi :Nilai Posterior :311937.56
Domba Tertembak
SKOR ANDA : 10
```

Gambar 6.7 Output Klasifikasi *Naïve Bayes*

- 7) Menentukan tingkat akurasi hasil Klasifikasi *Naïve Bayes* dengan rumus **Persamaan (6.3)** dibawah ini :

$$Akurasi = \frac{Total\ data - Data\ tidak\ sesuai}{Total\ data} \times 100\% \quad (6.3)$$

- 8) Kesimpulan.

Pada penelitian ini, pengujian dilakukan sebanyak 12 kali dengan total tembakan sebanyak 44 kali. Adapun jenis pengujian yang dilakukan yakni :

- 1) Mode Babi 4 Tembakan : Semua *Sillhouette* ditembak, ketika *Sillhouette* babi tertembak maka skor akan bertambah.
- 2) Mode Babi 3 Tembakan A : *Sillhouette* Ayam, Kalkun dan Domba ditembak tidak akan menambah skor. Begitu pula ketika tembakan ke-4 menembak *Sillhouette* Babi, skor tetap tidak bertambah.
- 3) Mode Babi 3 Tembakan B : *Sillhouette* Kalkun dan Domba ditembak tidak akan menambah skor, namun ketika peluru ke-3 ditembak ke arah *Sillhouette* Babi, maka skor akan bertambah.
- 4) Mode Ayam 4 Tembakan : Semua *Sillhouette* ditembak, ketika *Sillhouette* Ayam tertembak maka skor akan bertambah.
- 5) Mode Ayam 3 Tembakan A : *Sillhouette* Kalkun, Domba dan Babi ditembak tidak akan menambah skor. Begitu pula ketika tembakan ke-4 menembak *Sillhouette* Ayam, skor tetap tidak bertambah.
- 6) Mode Ayam 3 Tembakan B : *Sillhouette* Domba dan Babi ditembak tidak akan menambah skor, namun ketika peluru ke-3 ditembak ke arah *Sillhouette* Ayam, maka skor akan bertambah.
- 7) Mode Kalkun 4 Tembakan : Semua *Sillhouette* ditembak, ketika *Sillhouette* Kalkun tertembak maka skor akan bertambah.
- 8) Mode Kalkun 3 Tembakan A : *Sillhouette* Domba, Babi dan Ayam ditembak tidak akan menambah skor. Begitu pula ketika tembakan ke-4 menembak *Sillhouette* Kalkun, skor tetap tidak bertambah.

- 9) Mode Kalkun 3 Tembakan B : *Silhouette* Babi dan Ayam ditembak tidak akan menambah skor, namun ketika peluru ke-3 ditembakkan ke arah *Silhouette* Kalkun, maka skor akan bertambah.
- 10) Mode Domba 4 Tembakan : Semua *Silhouette* ditembak, ketika *Silhouette* Domba tertembak maka skor akan bertambah.
- 11) Mode Domba 3 Tembakan A : *Silhouette* Babi, Ayam dan Kalkun ditembak tidak akan menambah skor. Begitu pula ketika tembakan ke-4 menembak *Silhouette* Domba, skor tetap tidak bertambah.
- 12) Mode Domba 3 Tembakan B : *Silhouette* Ayam dan Kalkun ditembak tidak akan menambah skor, namun ketika peluru ke-3 ditembakkan ke arah *Silhouette* Domba, maka skor akan bertambah.

6.4.3 Hasil dan Analisis Pengujian

Hasil pengujian akurasi hasil klasifikasi *Naïve Bayes* ditunjukkan pada beberapa Tabel.

- 1) Mode Babi 4 Tembakan

Tabel 6.4 Hasil Pengujian Akurasi Klasifikasi *Naïve Bayes* Mode Babi 4 Tembakan

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
1	28,70	-21,43	11,78	41,12	Ayam Tertembak	Ayam Tertembak	0	Sesuai
2	38,80	-22,34	3,14	34,67	Kalkun Tertembak	Kalkun Tertembak	0	Sesuai
3	62,30	-30,68	-2,19	-11,99	Domba Tertembak	Domba Tertembak	0	Sesuai
4	48,10	14,10	24,95	-26,31	Babi Tertembak	Babi Tertembak	10	Sesuai

2) Mode Babi 3 Tembakan A

Tabel 6.5 Hasil Pengujian Akurasi Klasifikasi *Naïve Bayes* Mode Babi 3 Tembakan A

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
1	28,70	-21,43	11,78	41,12	Ayam Tertembak	Ayam Tertembak	0	Sesuai
2	38,80	-22,34	3,14	34,67	Kalkun Tertembak	Kalkun Tertembak	0	Sesuai
3	62,30	-30,68	-2,19	-11,99	Domba Tertembak	Domba Tertembak	0	Sesuai
4	48,10	14,10	24,95	-26,31	Babi Tertembak	-	0	Sesuai

3) Mode Babi 3 Tembakan B

Tabel 6.6 Hasil Pengujian Akurasi Klasifikasi *Naïve Bayes* Mode Babi 3 Tembakan B

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
1	38,80	-22,34	3,14	34,67	Kalkun Tertembak	Kalkun Tertembak	0	Sesuai
2	62,30	-30,68	-2,19	-11,99	Domba Tertembak	Domba Tertembak	0	Sesuai
3	48,10	14,10	24,95	-26,31	Babi Tertembak	Babi Tertembak	10	Sesuai

4) Mode Ayam 4 Tembakan

Tabel 6.7 Hasil Pengujian Akurasi Klasifikasi *Naïve Bayes* Mode Ayam 4 Tembakan

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
1	38,80	-22,34	3,14	34,67	Kalkun Tertembak	Kalkun Tertembak	0	Sesuai

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
2	62,30	-30,68	-2,19	-11,99	Domba Tertembak	Domba Tertembak	0	Sesuai
3	48,10	14,10	24,95	-26,31	Babi Tertembak	Babi Tertembak	0	Sesuai
4	28,70	-21,43	11,78	41,12	Ayam Tertembak	Ayam Tertembak	10	Sesuai

5) Mode Ayam 3 Tembakan A

Tabel 6.8 Hasil Pengujian Akurasi Klasifikasi *Naïve Bayes* Mode Ayam 3 Tembakan A

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
1	38,80	-22,34	3,14	34,67	Kalkun Tertembak	Kalkun Tertembak	0	Sesuai
2	62,30	-30,68	-2,19	-11,99	Domba Tertembak	Domba Tertembak	0	Sesuai
3	48,10	14,10	24,95	-26,31	Babi Tertembak	Babi Tertembak	0	Sesuai
4	28,70	-21,43	11,78	41,12	Ayam Tertembak	-	0	Sesuai

6) Mode Ayam 3 Tembakan B

Tabel 6.9 Hasil Pengujian Akurasi Klasifikasi *Naïve Bayes* Mode Ayam 3 Tembakan B

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
1	62,30	-30,68	-2,19	-11,99	Domba Tertembak	Domba Tertembak	0	Sesuai
2	48,10	14,10	24,95	-26,31	Babi Tertembak	Babi Tertembak	0	Sesuai
3	28,70	-21,43	11,78	41,12	Ayam Tertembak	Ayam Tertembak	10	Sesuai

7) Mode Kalkun 4 Tembakan

Tabel 6.10 Hasil Pengujian Akurasi Klasifikasi *Naïve Bayes* Mode Kalkun 4 Tembakan

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
1	62,30	-30,68	-2,19	-11,99	Domba Tertembak	Domba Tertembak	0	Sesuai
2	48,10	14,10	24,95	-26,31	Babi Tertembak	Babi Tertembak	0	Sesuai
3	28,70	-21,43	11,78	41,12	Ayam Tertembak	Ayam Tertembak	0	Sesuai
4	38,80	-22,34	3,14	34,67	Kalkun Tertembak	Kalkun Tertembak	10	Sesuai

8) Mode Kalkun 3 Tembakan A

Tabel 6.11 Hasil Pengujian Akurasi Klasifikasi *Naïve Bayes* Mode Kalkun 3 Tembakan A

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
1	62,30	-30,68	-2,19	-11,99	Domba Tertembak	Domba Tertembak	0	Sesuai
2	48,10	14,10	24,95	-26,31	Babi Tertembak	Babi Tertembak	0	Sesuai
3	28,70	-21,43	11,78	41,12	Ayam Tertembak	Ayam Tertembak	0	Sesuai
4	38,80	-22,34	3,14	34,67	Kalkun Tertembak	-	0	Sesuai

9) Mode Kalkun 3 Tembakan B

Tabel 6.12 Hasil Pengujian Akurasi Klasifikasi *Naïve Bayes* Mode Kalkun 3 Tembakan B

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
1	48,10	14,10	24,95	-26,31	Babi Tertembak	Babi Tertembak	0	Sesuai
2	28,70	-21,43	11,78	41,12	Ayam Tertembak	Ayam Tertembak	0	Sesuai
3	38,80	-22,34	3,14	34,67	Kalkun Tertembak	Kalkun Tertembak	10	Sesuai

10) Mode Domba 4 Tembakan

Tabel 6.13 Hasil Pengujian Akurasi Klasifikasi *Naïve Bayes* Mode Domba 4 Tembakan

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
1	48,10	14,10	24,95	-26,31	Babi Tertembak	Babi Tertembak	0	Sesuai
2	28,70	-21,43	11,78	41,12	Ayam Tertembak	Ayam Tertembak	0	Sesuai
3	38,80	-22,34	3,14	34,67	Kalkun Tertembak	Kalkun Tertembak	0	Sesuai
4	62,30	-30,68	-2,19	-11,99	Domba Tertembak	Domba Tertembak	10	Sesuai

11) Mode Domba 3 Tembakan A

Tabel 6.14 Hasil Pengujian Akurasi Klasifikasi *Naïve Bayes* Mode Domba 3 Tembakan A

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
1	48,10	14,10	24,95	-26,31	Babi Tertembak	Babi Tertembak	0	Sesuai

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
2	28,70	-21,43	11,78	41,12	Ayam Tertembak	Ayam Tertembak	0	Sesuai
3	38,80	-22,34	3,14	34,67	Kalkun Tertembak	Kalkun Tertembak	0	Sesuai
4	62,30	-30,68	-2,19	-11,99	Domba Tertembak	-	0	Sesuai

12) Mode Domba 3 Tembakan B

Tabel 6.15 Hasil Pengujian Akurasi Klasifikasi *Naïve Bayes* Mode Domba 3 Tembakan B

Tembakan ke-	Berat	X	Y	Z	Kelas	Hasil Sistem	Skor	Kesesuaian
1	28,70	-21,43	11,78	41,12	Ayam Tertembak	Ayam Tertembak	0	Sesuai
2	38,80	-22,34	3,14	34,67	Kalkun Tertembak	Kalkun Tertembak	0	Sesuai
3	62,30	-30,68	-2,19	-11,99	Domba Tertembak	Domba Tertembak	10	Sesuai

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 6.4 terlihat bahwa dari jumlah 44 data semuanya sesuai. Sehingga akurasi yang diperoleh *Scoring System* Otomatis Pada Lomba Menembak Dengan Target *Sillhouette* Hewan menggunakan Metode *Naive Bayes* ini berdasarkan **Persamaan (6.3)** adalah sebesar 100%. Proses perhitungan akurasi sebagai berikut :

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Total data} - \text{Data tidak sesuai}}{\text{Total data}} \times 100\% \\
 &= \frac{44 - 0}{44} \times 100\% \\
 &= \frac{44}{44} \times 100\% \\
 &= 100\%
 \end{aligned}$$

BAB 7 PENUTUP

Bab ini memuat penarikan kesimpulan berdasarkan tahap-tahap yang telah dikerjakan sebelumnya. Selain itu pada bab ini pula peneliti menyampaikan saran yang diharapkan dapat digunakan untuk pengembangan penelitian yang serupa dengan penelitian ini selanjutnya.

7.1 Kesimpulan

Sesuai dengan rumusan masalah yang diajukan diawal penelitian serta berdasarkan hasil analisis dari pengujian yang dilakukan maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Proses perancangan *Scoring System* Otomatis Pada Lomba Menembak Dengan Target *Sillhouette* Hewan dengan menyambungkan *Load Cell*, sensor *ADXL 335 Accelerometer*, dan *Limit Switch* ke kontroler input (Arduino Mega 2560) dan menghubungkan kontroler input (Arduino Mega 2560) ke pusat kontroler (Arduino UNO) menjadi sebuah sistem yang saling terhubung satu sama lain. Proses akuisisi pada *Load Cell* meliputi pembacaan berat pada *Sillhouette*. Akuisisi pada sensor *ADXL 335 Accelerometer* meliputi pembacaan percepatan sumbu gerak pada *Sillhouette*. Sedangkan proses akuisisi data pada *Limit Switch* meliputi pembacaan value ketika *Sillhouette* dalam keadaan normal dan tertembak.
2. Pada penelitian ini telah dibuat "*Scoring System* Otomatis Pada Lomba Menembak Dengan Target *Sillhouette* Hewan Menggunakan Metode *Naïve Bayes*". Dimana baik semua komponen alat yang digunakan maupun metode *Naïve Bayes* yang diterapkan dapat berjalan sesuai dengan yang diinginkan, terbukti dengan sistem dapat mengklasifikasikan jenis *Sillhouette* yang tertembak.
3. Akurasi yang diperoleh "*Scoring System* Otomatis Pada Lomba Menembak Dengan Target *Sillhouette* Hewan Menggunakan Metode *Naïve Bayes*"

yang diuji dengan jumlah data uji yakni sebanyak 44 data adalah senilai 100%.

7.2 Saran

Berdasarkan pembuatan dan hasil pada pengujian “*Scoring System* Otomatis Pada Lomba Menembak Dengan Target *Sillhouette* Hewan Menggunakan Metode *Naïve Bayes*”, dapat diberikan saran untuk pengembangan penelitian selanjutnya, antara lain:

1. Sistem dapat dikembangkan kedalam bentuk *Internet Of Things* (IoT).
2. Sistem dapat menggunakan wireless dalam proses pengiriman data.
3. Skor dapat diinterpretasikan kedalam papan skor berbasis led.





DAFTAR PUSTAKA

- Analog Device. (2010). *ADXL335*. Retrieved from Analog Device:
<http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL335.pdf>
- Arduino. (2018, May 21). *Arduino Uno Atmega328P*. Retrieved from Arduino:
<https://www.arduino.cc/en/Main/ArduinoBoardUno>
- Instruments, T. (2017, September). CD405xB CMOS Single 8-Channel Analog Multiplexer/Demultiplexer. Retrieved from
<http://www.ti.com/lit/ds/symlink/cd4051b.pdf>
- Invensense. (2011). *MPU-6050 / TDK*. Retrieved from Invensense:
<https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>
- JEHACESHOOTINGCLUB. (2017, May 13). Peraturan Pertandingan/Lomba Menembak. Kabupaten Bekasi, Jawa Barat, Indonesia.
- LenteraKecil. (2016, February 18). *Manfaat dan Falsafah Olahraga Menembak*. Retrieved from Lentera Kecil: <https://lenterakecil.com/manfaat-dan-falsafah-olahraga-menembak/>
- Mäuselein, S., Mack, O., & Schwartz, R. (2009). Investigations into the use of singlecrystalline silicon as mechanical spring in load cells.
- Muliadi, R. (2014). ALAT SCORING BOARD PERTANDINGAN BOLA BASKET BERBASIS MIKROKONTROLER AT89S51. *Sport Science*, 39-46.
- Norhabibah, S., Andhyka, W., & Risqiwati, D. (2016). Rancang Bangun Sistem Monitoring Deteksi Jatuh untuk Manula dengan Menggunakan Accelerometer . 43 - 52.
- Prateek Gupta, D. D. (2017). Gesture Controlled Mobile Robotic Arm Vehicle Using Accelerometer. *International Research Journal of Engineering and Technology (IRJET)* , 1245.
- Ramadhan, H., Maulana, R., & Ichsan, M. H. (2018). Scoring System Otomatis pada Lomba Menembak dengan Target Silhouette Hewan Menggunakan Logika Fuzzy.
- Rivani, F., Syauqy, D., & Ichsan, M. H. (2017). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. *Implementasi Naive Bayes pada Embedded System untuk Menentukan Status Gizi Bayi*, 1277-1284.
- Wardhani, K. F. (2005). Sistem Pendukung Keputusan Kelayakan Kredit Pemilikan Mobil dengan Metode Scoring System. Surabaya : STIKOM.
- Xtopher. (2016, July 22). *Load Cell Amplifier - HX711*. Retrieved from Sparkfun:
<https://www.sparkfun.com/news/2150>

